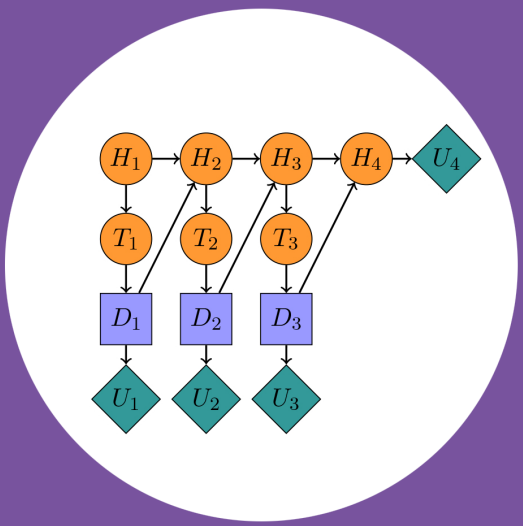
Department of Mathematics and Systems Analysis

Mixed-integer formulations for large-scale energy-environmental optimization

Olli Herrala



Aalto University

DOCTORAL THESES

Aalto University publication series
DOCTORAL THESES 158/2024

Mixed-integer formulations for large-scale energy-environmental optimization

Olli Herrala

A doctoral thesis completed for the degree of Doctor of Science (Technology) to be defended, with the permission of the Aalto University School of Science, at a public examination held at the lecture hall H304 (Otakaari 1, Espoo, Finland) of the school on 6 September 2024 at 12 noon.

Aalto University
School of Science
Department of Mathematics and Systems Analysis
Gamma-opt research group

Supervising professor
Professor Fabricio Oliveira, Aalto University, Finland

Thesis advisor
Research Professor Tommi Ekholm, Finnish Meteorological Institute, Finland

Preliminary examiners
Professor Martine Labbé, Université libre de Bruxelles, Belgium

Professor Giovanni Pantuso, University of Copenhagen, Denmark

Opponent
Professor Giovanni Pantuso, University of Copenhagen, Denmark

Aalto University publication series
DOCTORAL THESES 158/2024

ISBN 978-952-64-1950-3 (printed)
ISBN 978-952-64-1951-0 (pdf)
ISSN 1799-4934 (printed)
ISSN 1799-4942 (pdf)
http://urn.fi/URN:ISBN:978-952-64-1951-0

Unigrafia Oy
Helsinki 2024

Finland

NORDIC SWAN ECOLABEL
Printed matter
4041-0619

Author
Olli Herrala

Name of the doctoral thesis
Mixed-integer formulations for large-scale energy-environmental optimization

Publisher School of Science

Unit Department of Mathematics and Systems Analysis

Series Aalto University publication series DOCTORAL THESES 158/2024

Field of research Systems and Operations Research

Manuscript submitted 22 July 2024 **Date of the defence** 6 September 2024

Permission for public defence granted (date) 17 June 2024 **Language** English

☐ **Monograph** ☒ **Article thesis** ☐ **Essay thesis**

Abstract

To support decision making in complex systems, different mathematical programming approaches have been developed for both modelling the decision process and finding the best decisions or policies. This dissertation focuses on decision making in the context of climate change mitigation, considering two different perspectives on the topic. The first is a global perspective of research and development of negative emission technologies and their effect on the optimal emission levels in the next 50 years. The second is a more localized perspective of regulating a competitive power market, aiming to efficiently reduce the emissions of electricity production.

The structures contained within these problems render them incompatible with existing solution methods. First, many of the uncertainties in climate change mitigation, such as the cost of reducing emissions in the future, depend on the earlier decisions such as the level of research investment. While different types of decision-dependent uncertainty have been researched before, combining these types in a way that would allow for accurate modeling of the decision process has not been possible. Second, a bilevel hierarchical structure of a power market with a transmission system operator and electricity producers has received significant attention in the literature, but these methods are not directly applicable to problems with a third hierarchical level representing, in our case, the international regulator.

This dissertation enables more realistic modeling of decision-dependent uncertainty and hierarchical decision making by reformulating the problems using mixed-integer programming (MIP). Using mixed-integer optimization as the main solution framework allows us to utilize the vast developments in solving mixed-integer models. Additionally, implementing explicit risk measures in MIP models has received significant attention in the literature, and these developments can be applied to the proposed models with minor adjustments. However, the computational efficiency of solving mixed-integer models depends not only on the solution method, but also the formulation used. The articles in this dissertation discuss and compare three different MIP formulations for limited memory influence diagrams, and two single-level reformulations for trilevel equilibrium models.

Despite their simplified nature, the case studies in this dissertation provide policy insights on cost-benefit optimal emission trajectories and the effect of a carbon tax on the Nordic electricity market. Such models can help justify decisions when developing policies in complex and controversial contexts such as climate change mitigation. While the focus of the dissertation is on energy and environment, the methodological developments in this dissertation are equally applicable to a variety of problems in fields such as healthcare, systems monitoring and traffic planning.

Keywords mixed-integer optimization, equilibrium modelling, climate change mitigation, energy systems modeling

ISBN (printed) 978-952-64-1950-3 **ISBN (pdf)** 978-952-64-1951-0

ISSN (printed) 1799-4934 **ISSN (pdf)** 1799-4942

Location of publisher Helsinki **Location of printing** Helsinki **Year** 2024

Pages 130 **urn** http://urn.fi/URN:ISBN:978-952-64-1951-0

Tekijä
Olli Herrala

Väitöskirjan nimi
Sekalukuoptimointimalleja energia- ja ympäristöpäätöksentekoon

Julkaisija Perustieteiden korkeakoulu

Yksikkö Matematiikan ja systeemianalyysin laitos

Sarja Aalto University publication series DOCTORAL THESES 158/2024

Tutkimusala Systeemi- ja operaatiotutkimus

Käsikirjoituksen pvm 22.07.2024 **Väitöspäivä** 06.09.2024

Väittelyluvan myöntämispäivä 17.06.2024 **Kieli** Englanti

☐ **Monografia** ☒ **Artikkeliväitöskirja** ☐ **Esseeväitöskirja**

Tiivistelmä

Monimutkaisten järjestelmien päätöksenteon tukemiseksi on kehitetty erilaisia matemaattisen optimoinnin menetelmiä sekä päätöksentekoprosessin mallintamiseen että parhaiden päätösten tai toimintatapojen löytämiseen. Tässä väitöskirjassa keskitytään ilmastonmuutoksen hillitsemiseen liittyvään päätöksentekoon kahdesta eri näkökulmasta. Ensimmäinen on maailmanlaajuinen näkökulma negatiivisten päästöjen teknologioiden tutkimukseen ja kehittämiseen sekä niiden vaikutukseen optimaalisiin päästötasoihin seuraavien 50 vuoden aikana. Toinen, paikallisempi näkökulma liittyy kilpailuun perustuvien sähkömarkkinoiden sääntelyyn, tavoitteena vähentää tehokkaasti sähköntuotannon päästöjä.

Nykyiset ratkaisumenetelmät eivät ole yhteensopivia näihin ongelmiin sisältyvien rakenteiden kanssa. Ensinnäkin monet ilmastonmuutoksen hillitsemiseen liittyvät epävarmuustekijät, kuten päästöjen vähentämisen kustannukset tulevaisuudessa, riippuvat aiemmista päätöksistä, kuten tutkimusinvestointien tasosta. Vaikka erilaisia päätösriippuvaisia epävarmuustyyppejä on tutkittu aiemminkin, niiden yhdistäminen ja näiden päätöksentekoprosessien tarkka mallintaminen ei ole ollut mahdollista. Toiseksi kirjallisuudessa on kiinnitetty paljon huomiota kaksitasoiseen hierarkiseen sähkömarkkinamalliin, jossa siirtoverkonhaltija ja sähköntuottajat tekevät päätöksiä toisistaan riippuen, mutta näitä menetelmiä ei voida suoraan soveltaa kolmitasoisiin ongelmiin, joissa mallinnetaan lisäksi esimerkiksi kansainvälistä sääntelyviranomaista.

Tämä väitöskirja mahdollistaa päätösriippuvaisen epävarmuuden ja hierarkkisen päätöksenteon realistisemman mallintamisen muotoilemalla ongelmat uudelleen käyttäen sekalukuoptimointia (mixed-integer optimization, MIP). Sekalukoptimoinnin soveltaminen mahdollistaa näiden mallien ratkaisemisessa tapahtuneen laajan kehityksen hyödyntämisen. Lisäksi kirjallisuudessa on kiinnitetty paljon huomiota eksplisiittisten riskimittojen toteuttamiseen MIP-malleissa, ja näitä voidaan soveltaa myös ehdotettuihin malleihin. Sekalukuoptimointimallien ratkaisemisen laskennallinen tehokkuus riippuu paitsi ratkaisumenetelmästä, myös käytetystä ongelman muotoilusta. Tässä väitöskirjassa käsitellään ja vertaillaan kolmea erilaista sekalukumallia rajoitetun muistin vaikutuskaavioille sekä kahta mallia kolmitasoisille tasapainomalleille. Yksinkertaistetusta luonteestaan huolimatta tämän väitöskirjan tapaustutkimukset tarjoavat näkemyksiä kustannus-hyöty-optimaalisista päästötasoista ja hiiliveron vaikutuksesta Pohjoismaiden sähkömarkkinoihin. Tällaiset mallit voivat auttaa perustelemaan päätöksiä, kun kehitetään politiikkaa monimutkaisissa ja kiistellyissä yhteyksissä, kuten ilmastonmuutoksen hillitsemisessä. Vaikka väitöskirjassa keskitytään energia- ja ympäristöpäätöksentekoon, siinä kehitettyjä menetelmiä voidaan soveltaa yhtä lailla erilaisiin ongelmiin terveydenhuollon, järjestelmien seurannan ja liikennesuunnittelun kaltaisilla aloilla.

Avainsanat sekalukuoptimointi, tasapainomallinnus, ilmastonmuutoksen hillintä, sähköjärjestelmämallinnus

ISBN (painettu) 978-952-64-1950-3 **ISBN (pdf)** 978-952-64-1951-0

ISSN (painettu) 1799-4934 **ISSN (pdf)** 1799-4942

Julkaisupaikka Helsinki **Painopaikka** Helsinki **Vuosi** 2024

Sivumäärä 130 **urn** http://urn.fi/URN:ISBN:978-952-64-1951-0

Preface

This dissertation is a result of four years of learning. It has not been an easy four years, but looking back, I can see both personal development and improvement in my research skills. This progress was made possible by a large number of people around me, and this is where I try to acknowledge at least most of them.

First, I want to thank my supervisor, Professor Fabricio Oliveira. You've been incredibly supportive and flexible even when my circumstances have been less than ideal. When I applied for an MSc thesis position in your group five years ago, I had no plan to pursue a doctoral degree. However, I obviously did not get enough of you during that thesis process and decided to stay for four more years. Honestly, I think that is one of the best decisions I've made. Thank you for your support, I hope we'll be able to collaborate in the future as well.

I've been lucky to have not one, but two very supportive academics guiding my dissertation process, and I want to also thank my advisor, Professor Tommi Ekholm. I remember starting my MSc thesis and knowing that whenever the models get too confusing, I can come and ask you for help. Thank you for all the feedback on my ideas and texts, it has always been very constructive and useful. And furthermore, thank you for introducing me to the Finnish Operations Research Society, I am honored to be able to follow in your footsteps as the president of the society.

Moreover, I have had the chance to collaborate with a number of great researchers during these years. Professor Steven Gabriel, thank you for the opportunity to work together on one of the papers in this dissertation, the writing process wasn't as straightforward as we initially thought, but the result was a good piece of research and it was a great experience working with you. Professor Philine Schiewe, thank you for the opportunity to work together on a paper outside this dissertation, it was a welcome break from my other research. My other co-authors: Helmi, Irene, Jaan and Topias, it was a pleasure working with all of you. Finally, I want to thank Professors Giovanni Pantuso and Martine Labbé for their kind and insightful pre-examination comments, and the Research Council of Finland for financially

supporting the thesis work.

In addition to working on my research, I've been able to work with students both as a thesis advisor and course assistant. I very much enjoyed these experiences, and I want to thank every student I had the privilege to teach, the joy in your eyes when you figured things out was always a highlight of my day, and I hope you will find use for the things you learned.

These years have not been just hard work and for that, I am grateful to all my friends. Everyone at the Systems Analysis Laboratory, thank you for all the lunch discussions, tea times, coffee machine chats, dog-sitting, and more. I also want to thank the Guild of Physics for the 10 years of fun in Otaniemi, I guess it's finally time for this old man to leave the guild room for good. Also, a big thank you to my dive buddies for giving me the opportunity to relax when life wasn't otherwise very relaxing.

We're approaching the end of the preface, but there are still very important people that must be thanked before we get there. My wife Liinu, thank you for staying with me throughout these years. It was always supposed to be you doing a doctoral degree, not me. However, life happened and here we are. Life will most likely continue to throw unexpected things our way, but I'm sure we can handle them together. I'm also grateful to my parents and my family, I would not be writing this dissertation without all the support I had throughout the last few decades.

Finally, I thank God for the peace and love that has carried me through these years. "Now may the Lord of peace himself give you peace at all times and in every way".

Espoo, July 22, 2024,

Olli Herrala

Contents

Preface 1

Contents 3

List of Publications 5

Author's Contribution 7

List of Figures 9

List of Tables 11

Abbreviations 13

1. Introduction 15
- 1.1 Background . . . 15
- 1.2 Representing and solving decision problems . . . 16
- 1.3 Climate change mitigation . . . 17
- 1.4 Dissertation objectives . . . 18

2. Methodological background 21
- 2.1 Mixed-integer optimization . . . 21
- 2.2 Influence diagrams . . . 24
  - 2.2.1 Decision Programming . . . 26
  - 2.2.2 Rooted junction trees . . . 27
- 2.3 Bilevel optimization . . . 29

3. Contributions 33
- 3.1 Paper I . . . 35
- 3.2 Paper II . . . 35
- 3.3 Paper III . . . 36
- 3.4 Paper IV . . . 36

4. Discussion 37

References 41

Publications 45

List of Publications

This thesis consists of an overview and of the following publications which are referred to in the text by their Roman numerals.

I Helmi Hankimaa, Olli Herrala, Fabricio Oliveira and Jaan Tollander de Balsch. Solving influence diagrams via efficient mixed-integer programming formulations and heuristics. Submitted to *INFORMS Journal on Computing*, December 2023.

II Olli Herrala, Topias Terho, Fabricio Oliveira. Risk-averse decision strategies for influence diagrams using rooted junction trees. Submitted to *Operations Research Letters*, December 2023.

III Olli Herrala, Tommi Ekholm, Fabricio Oliveira. Solving decision problems with endogenous uncertainty and conditional information revelation using influence diagrams. Submitted to *Omega*, February 2024.

IV Olli Herrala, Steven A. Gabriel, Fabricio Oliveira, Tommi Ekholm. A novel strong duality -based reformulation for trilevel infrastructure models in energy systems development. *Journal of the Operational Research Society*, July 2024.

Author's Contribution

Publication I: "Solving influence diagrams via efficient mixed-integer programming formulations and heuristics"

Oliveira and Herrala collaborated on writing the paper. Hankimaa and Tollander implemented the software package with supervision and collaboration from Herrala. Hankimaa implemented the case study in a BSc thesis with supervision from Oliveira and Herrala.

Publication II: "Risk-averse decision strategies for influence diagrams using rooted junction trees"

Herrala is the primary author. He proposed the topic and implemented the models and wrote the manuscript together with Terho. Oliveira helped finalize the manuscript.

Publication III: "Solving decision problems with endogenous uncertainty and conditional information revelation using influence diagrams"

Herrala is the primary author. He wrote most of the manuscript. Ekholm proposed and parametrized the case study, and wrote the description of the case study. Herrala formalized the methodological contributions together with Oliveira and implemented the case study.

Publication IV: "A novel strong duality -based reformulation for trilevel infrastructure models in energy systems development"

Herrala is the primary author. Gabriel proposed the topic and all authors designed the energy system model together. Herrala designed the novel formulation, implemented both formulations and wrote most of the manuscript.

List of Figures

2.1 An LP feasible region and two inequalities exposing x_{IP} . 22
2.2 The influence diagram of the pig farm problem. 24
2.3 The gradual rooted junction tree of the pig farm problem. 28

List of Tables

3.1 Summary of publications 34

Abbreviations

CO2 Carbon dioxide

CVaR Conditional value-at-risk

ID Influence diagram

LCP Linear complementarity problem

LIMID Limited memory influence diagram

MI(L)P Mixed-integer (linear) programming

MPEC Mathematical programming with equilibrium constraints

MPPDC Mathematical programming with primal and dual constraints

RJT Rooted junction tree

1. Introduction

1.1 Background

When making decisions, inaccurate information about the consequences can lead to unfavorable outcomes. Alas, our world is and has always been full of uncertainty. When you leave your house, there is a chance it will start raining at some point; investment strategies are based on possibly inaccurate models of how the value of stocks will develop in the future; and even the scientific community faces challenges when discussing climate change because we are uncertain about its effects. Despite all this uncertainty, we still need to make decisions. Should you take an umbrella with you in the morning? What combination of stocks is the most likely to result in high profits with low risk? What should the global CO2 emission target levels be in the future?

Sun Tzu wrote, "Thus a victorious army wins its victories before seeking battle; an army destined to defeat fights in the hope of winning." While this dissertation does not further discuss military tactics, this quote powerfully describes the more general idea of decision making under uncertainty, as careful planning of a decision strategy usually results in better outcomes than making decisions without considering the possible futures (Birge, 1982). For example, in climate change mitigation, it is imperative to factor in uncertainties by choosing actions that position us favorably in probable future scenarios while safeguarding against catastrophic outcomes in less likely scenarios. As our understanding of climate dynamics evolves, these strategies should be adjusted accordingly to ensure both short- and long-term benefits.

The theory of decision-making under uncertainty started with sequential decisions under uncertainty (Wald, 1947; Dvoretzky et al., 1952), and shortly after, dynamic programming (Bellman, 1954) was introduced as an approach to solve such problems. The following year, *Linear Programming Under Uncertainty* (Dantzig, 1955) was published, eventually leading to

many of the fundamental developments making the models in, e.g., this dissertation possible to solve. Simultaneously, Von Neumann and Morgenstern (1947) introduced the field of game theory for modeling problems with more than one decision maker.

1.2 Representing and solving decision problems

The philosophy and methodology of forming quantitative models to represent decision problems is referred to as *decision analysis*, considering the interplay between decisions, uncertainty, and outcomes while seeking a decision strategy that maximizes some form of utility[1]. Two well-known visual representations in decision analysis are decision trees and influence diagrams (discussed in detail in Section 2.2). The usefulness of influence diagrams lies in the idea that they are "at once both a formal description of the problem that can be treated by computers and a representation easily understood by people in all walks of life and degrees of technical proficiency" (Howard and Matheson, 2005).

This dissertation discusses a number of linear, mixed-integer, and equilibrium models, all of which fall under the modeling paradigm of *mathematical programming*. In mathematical programming, a real-world decision problem is expressed in a language that allows one to use optimization methods to solve the problem. The decisions are represented as variables, and the solution quality is measured with an *objective function* depending on the decision variables and uncertainty realizations. Finally, the *constraints* of the problem are relations such as inequalities and equalities defining the *feasible* values of the decision variables.

The language of mathematical programming allows not only efficient communication of models but also the development of efficient solution methods for these problems. For instance, the Simplex method (Dantzig et al., 1954) can solve problems with real-valued decision variables, a linear objective function, and constraints represented with affine inequalities or equalities. A particular class of problems in this dissertation turns out to be very close to this linear formulation: if some of the decision variables are instead allowed to take only integer values, the resulting mixed-integer linear problem can be solved using, e.g., branch-and-bound methods (Land and Doig, 1960). The details of formulating and solving these *mixed-integer programming* models are discussed in Section 2.1.

[1] In practice, probabilities and utility functions can be subjective, and obtaining them from decision makers can thus be difficult. However, in this dissertation, we assume that these necessary elements of the model are available and instead focus on the representations of the decision problem.

1.3 Climate change mitigation

As discussed earlier, decision making under uncertainty can be as simple as choosing whether or not to take an umbrella with you in the morning, but it can also be as complex and global as climate change mitigation. The scientific discourse on the latter began in 1979 in Geneva, Switzerland, where climate experts gathered to discuss the possibility that human action is changing the global climate (World Climate Conference, 1979). Today, we understand the mechanisms of climate change significantly better (Lee et al., 2023) and have set target limits for global temperature increase (UNFCCC, 2015). Despite this increased understanding, there is still considerable uncertainty in some of the underlying mechanisms, making it challenging to create efficient policies for limiting global warming. Of course, immediately building enormous amounts of solar or wind power generation capacity would help reduce emissions from electricity production, but it would also result in challenges with raw material availability, energy system stability, and overall costs. On the other hand, doing nothing would result in significant climate damages (Lee et al., 2023), suggesting that some level of emission reduction is beneficial.

Nordhaus (1991) first discussed greenhouse gas abatement cost-benefit analysis, with the goal of maximizing overall net economic welfare. The analysis is based on formulating a greenhouse damage function and an abatement cost function, then finding an abatement level that minimizes the total cost of abatement and damages (i.e., maximizes overall welfare). Similar analysis is used today, with improvements to the aforementioned functions and the overall structure of the model. A model of particular interest in this dissertation is SCORE (Stochastic Cost Optimization for Reducing Emissions) (Ekholm, 2014), an integrated assessment model where abatement decisions and uncertainty revelation happen gradually. This gradual learning in the model results in a more realistic analysis where the abatement strategies can adapt after more information is revealed, compared to similar models without learning. However, this adds a level of complexity to the model in the form of uncertainty, and determining the probability structure of this learning process is a significant challenge in itself.

Even if we had perfect information about the climate dynamics and could determine the most efficient trajectories for emission reduction, reaching these emission targets requires changes in the behavior of both energy consumers and producers, posing an entirely different problem that must be tackled from a more local energy market level. For this reason, different incentives for emissions reduction have been considered. A relatively straightforward incentive is carbon tax (Köppl and Schratzenstaller, 2023) that energy producers pay according to the amount of emissions produced in their operation. However, in a connected market where multiple pro-

ducers compete in different countries, and the producers are assumed to maximize their profit with no interest in decreasing emissions, such incentives might have unexpected effects on the system. For example, a carbon tax might simply result in higher prices for consumers with little impact on production or increased imports from countries with a lower carbon tax (Lin and Li, 2011). These market equilibrium problems are commonly modeled using bilevel optimization, discussed in Section 2.3 of this dissertation.

1.4 Dissertation objectives

This dissertation is motivated by challenging decision problems stemming from climate change mitigation, considering two applications with different methodological contributions. While the focus is on climate change mitigation, the formulations in this dissertation can be utilized in a wide range of applications. Methodologically, the overall goal of this dissertation is to develop and improve solution methods for limited memory influence diagrams and hierarchical decision making. The main application is a large-scale climate research and development planning problem extending the SCORE model (Ekholm, 2014) to consider investments into climate research. The assumption is that research efforts can provide better information about the dynamics underlying climate change, and thus enable more informed decisions on the emission targets.

To efficiently model the decision-dependent or *endogenous* effect of climate research, a large part of this dissertation is devoted to influence diagrams and how to convert this intuitive representation into an efficient mathematical optimization formulation. Paper I discusses and improves Decision Programming (originally presented in Salo et al., 2022) and Paper II extends the rooted junction tree models originally presented in Parmentier et al. (2020). In addition to computational improvements, Paper I also introduces a Julia (Bezanson et al., 2017) interface for creating and solving (limited memory) influence diagrams, increasing the potential impact of the method as users no longer need substantial knowledge about MIP models to solve their influence diagrams.

While influence diagrams offer an intuitive way to represent decision-dependent probability distributions, learning the climate dynamics represents a second type of decision-dependent uncertainty. The formulations in Papers I and II are further generalized in Paper III by introducing "Type 2" endogenous uncertainty (Apap and Grossmann, 2017; Hellemo et al., 2018), specifically conditional information revelation, allowing us to solve the large-scale climate change mitigation problem in Paper III. Overall, these developments result in an efficient, generally applicable solution framework for decision problems represented as influence diagrams.

The global CO2 target levels obtained in Paper III stem from a high-level perspective of climate change, while in reality, reaching these targets requires the cooperation of different decision makers with competing interests. In order to model this setting, Paper IV considers a trilevel perspective of power markets where electricity producers only consider operational profit and incentives such as a carbon tax (Köppl and Schratzenstaller, 2023) are required for them to decrease their emissions. Taking this competitive aspect into consideration allows for investigating the effectiveness of different incentives in a realistic, hierarchical system. To solve these market models, Paper IV presents a novel single-level reformulation for trilevel optimization problems, aiming to improve computational efficiency compared to the reformulation presented in Gabriel et al. (2022). The two reformulations are compared using a case study representing the electricity market in the Nordic countries. Overall, this dissertation considers two connected perspectives in climate change mitigation, providing effective solution methods for the problems arising in an energy system and a global level of reducing greenhouse gas emissions.

2. Methodological background

In mathematical programming, the goal is to find values for decision variables such that the constraints of the problem are satisfied and the objective function value is minimized (or maximized). A general form of an optimization problem is

$$\text{min. } f(x) \tag{2.1}$$

$$\text{s.t. } g_i(x) \leq 0,\ \forall i \in \{1, ..., m\} \tag{2.2}$$

$$h_j(x) = 0,\ \forall j \in \{1, ..., l\} \tag{2.3}$$

$$x \in X, \tag{2.4}$$

where f, g_i and h_j are functions mapping the values of the decision variables x to $\mathbb{R}$. The nature of these functions as well as the set X determines the complexity of finding the optimal solution (or even a feasible solution). A particularly notable example is *linear programming* (LP) where f is linear, and g and h are affine. Often, for linear problems, we assume X to be the nonnegative orthant. Such problems can be efficiently solved using, e.g., Simplex or Barrier methods (Dantzig et al., 1954; Mehrotra, 1992).

This chapter starts by introducing mixed-integer optimization, a special case of (2.1)-(2.4), and the relevant solution methods for such problems. Section 2.2 continues with a description of influence diagrams and two MIP reformulations for finding optimal strategies for an influence diagram. Finally, Section 2.3 discusses the *optimality conditions* of the problem (2.1)-(2.4) and how they are used in bilevel optimization.

2.1 Mixed-integer optimization

When modeling real-world decision problems, one soon notices that many decisions are discrete instead of continuous. In *mixed-integer linear programming* (MILP), the set X contains a restriction that some or all of the variables x are integer-valued, and the functions f, g and h are affine.

Enforcing integrality not only allows for more realistic modeling of discrete phenomena but also enables representing logical relationships (Raman and Grossmann, 1994) within the model. During the last few decades, efficient methods for solving such problems have been developed, making these techniques meaningful in practice. In contrast, nonlinear functions f, g or h would result in mixed-integer nonlinear programming (MINLP) (Lee and Leyffer, 2011), a considerably harder class of problems with less prominent solution methods. In this dissertation and a significant part of optimization literature (e.g., Wolsey, 2020), the term *mixed-integer programming* (MIP) is used interchangeably with MILP.

Solving MIP problems is based on the idea of solving linear programming (LP) relaxations (that is, removing the integrality constraints from the formulation) of the original problem and iteratively adding constraints, removing non-integer solutions until the convex hull of the integer feasible region is correctly approximated at the integer optimal solution. The basic idea of such constraints is presented in Fig. 2.1, where an LP optimal point x_{LP} is "cut" out of the problem by adding two affine constraints, exposing the true optimum x_{IP}. With these constraints, the LP relaxation solution becomes x_{IP}, and the solution process terminates with the optimal solution. Solution methods based solely on adding new constraints to the MIP are referred to as *cutting plane* methods, using, e.g., Gomory fractional cuts (Gomory, 1958) and lift-and-project cuts (Lovász and Schrijver, 1991). However, these solution methods are considered ineffective and prone to numerical issues.

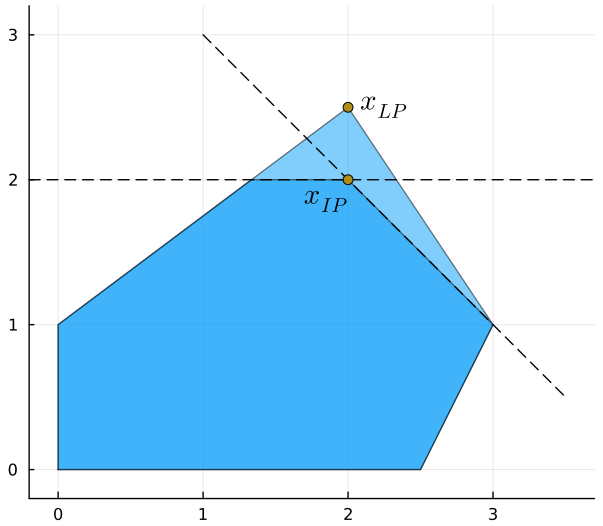


Figure 2.1. An LP feasible region and two inequalities exposing x_{IP}

This idea of iteratively adding constraints approximating the convex hull of the integer feasible region X leads to a discussion on good MIP

formulations. If the LP relaxation of the problem is exactly the convex hull of the integer feasible region, no additional constraints are required as the initial relaxation solution is guaranteed to be optimal for the MIP. This is the case for problems $\min\{c^\top x : \ Ax \leq b, \ x \in \mathbb{Z}^n_+\}$ with totally unimodular A and integral b (Wolsey, 2020). Notably, such problems include maximum flow or minimum cost flow problems where we consider a network of nodes. However, most MIP problems do not have this structure and thus require additional constraints to describe the convex hull. For such problems, it is important to use formulations that approximate the convex hull as closely as possible.[1] Notably, Papers I and II in this dissertation compare different MIP formulations corresponding to influence diagrams, discussed in Section 2.2.

The most prominent alternative for cutting-plane methods is branch-and-bound (Land and Doig, 1960), a divide-and-conquer approach where the original LP relaxation is recursively split into subproblems. For example, the LP optimal solution to the problem in Fig. 2.1 is $x_{LP} = (2, 2.5)$, where $x_2 = 2.5$ is fractional. The first step of branch-and-bound would be to solve two subproblems corresponding to the original LP relaxation with an additional constraint $x_2 \leq 2$ or $x_2 \geq 3$. Each integer feasible point in the original problem is in exactly one of these MIP subproblems, and thus, the better integer optimal point between these two subproblems is the optimal solution to the original MIP problem.

When these two subproblems are solved, whenever we obtain a fractional LP relaxation solution, we either create new subproblems (branching) or prune the branch if the LP solution value is worse than the best integer solution found so far (bounding). Once all branches have been explored or pruned, the best integer solution found is the optimal solution to the MIP problem. Similarly to the pure cutting plane approach, branch-and-bound can be computationally inefficient.

Despite the practical issues with both branch-and-bound and cutting planes, they are widely used today for solving MIPs. The key observation was made by Padberg and Rinaldi (1991): combining the two approaches by adding cutting planes during a branch-and-bound search greatly improves the computational efficiency of solving MIPs. This approach is known as branch-and-cut and is the cornerstone of modern MIP solvers, finding globally optimal solutions efficiently. These developments, combined with the versatility of MIP in modifying the model with additional constraints, are the main reason for utilizing MIPs in this dissertation. To make the solvers even more efficient is a continuing line of research, as companies like Gurobi (Gurobi Optimization, LLC, 2023), FICO (Fair Isaac Corporation, 2024) and IBM (Cplex, IBM ILOG, 2022) aim to have the fastest

[1]The formulations should also have a small number of constraints to avoid computational challenges; in practice, one might have to balance between the volume of the LP feasible region and the number of constraints.

solver in the market.

Another key component of an efficient MIP solver is heuristics (Berthold, 2006). While cutting planes can be useful in approaching the optimal solution "from the outside", the bounding part of branch-and-bound relies on finding good integer solutions. The goal of heuristics is to find good integer solutions by, e.g., using knowledge about the problem structure in constructing a feasible solution (constructive heuristics, e.g., Petch and Salhi (2003)) or starting with an integer solution found by the solver and trying to improve it by making small changes to the variable values (improvement heuristics, e.g., Van Breedam (1995)).

2.2 Influence diagrams

An influence diagram (ID) $G(N, A)$ is a directed acyclic graph with directed arcs $a \in A = \{(i, j) \mid i, j \in N\}$ connecting nodes $j \in N$. In Fig. 2.2, the influence diagram representation of the pig farm problem from Lauritzen and Nilsson (2001) is presented. The circular *chance nodes* $j \in N^C \subseteq N$ represent uncertain events, and the square *decision nodes* $j \in N^D \subseteq N$ represent decisions. Finally, the diamond-shaped[2] *value nodes* $j \in N^V \subseteq N$ represent consequences associated with the decisions and chance event realizations.

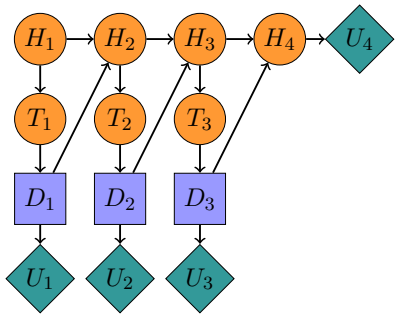


Figure 2.2. The influence diagram of the pig farm problem.

In the visual representation of an ID, the arrows between nodes represent the arcs $a \in A$ that, in turn, represent influence between the nodes in an intuitive way. We define the information set of a node $j \in N$ as the set of its immediate predecessors $I(j) = \{i \in N \mid (i, j) \in A\}$. We assume each decision and chance node to have a discrete, finite set of states $s_j \in S_j$. The probability distribution over the states $s_j \in S_j$ for a chance node $j \in N^C$ is conditional on the states of the nodes in the information set $I(j)$. We denote

[2]Some authors use hexagonal value nodes instead.

this *information state* as $s_{I(j)} \in S_{I(j)}$, and the realization of the random event as $X_j \in S_j$. Therefore, the (conditional) probability distribution associated with a chance node j is denoted as $\mathbb{P}(X_j = s_j \mid X_{I(j)} = s_{I(j)})$. The value nodes $v \in N^V$ represent utility functions $U_v : S_{I(v)} \to \mathbb{R}$, and while we assume these to be functions mapping each information state to exactly one utility value, some references (e.g., Parmentier et al., 2020) instead discuss stochastic value nodes. We note that this is not a significant difference, as we can instead use a deterministic value node v and introduce an additional chance node $j \in I(v)$ representing the stochasticity.

For decision nodes $j \in N^D$, the incoming arcs (and thus, the information set) specify the information available to the decision maker at the time of the decision. For example, in Fig. 2.2, the decision in D_1 is made knowing the outcome of T_1, but not H_1. We define a local decision strategy $Z_j : S_{I(j)} \to S_j$ so that when observing $s_{I(j)}$ results in choosing the decision alternative s_j, we have $Z_j(s_{I(j)}) = s_j$. This can be written using an indicator function $\mathbb{I} : S_{I(j)} \times S_j \to \{0, 1\}$ such that $\mathbb{I}(s_{I(j)}, s_j) = 1$ if and only if $Z_j(s_{I(j)}) = s_j$. We note that this is a somewhat restrictive approach precluding mixed strategies where an arbitrary number of decision alternatives $s_j \in S_j$ would have a nonzero probability of being chosen given a single information state $s_{I(j)}$, but this is a necessary restriction for the models used in the dissertation. We will next discuss how to solve influence diagrams, that is, find strategies $Z \in \mathbb{Z}$ maximizing (or minimizing) an objective function such as expected utility.

Often, influence diagrams can be solved using decision trees or by reducing the diagram representation as discussed in Howard and Matheson (2005). However, if the *no-forgetting* assumption, stating that the previous information must be available to the decision maker when making a decision, does not hold, these approaches can not be used. To circumvent this limitation, Lauritzen and Nilsson (2001) introduce limited memory influence diagrams (LIMID). The disadvantage of LIMIDs compared to influence diagrams satisfying the no-forgetting assumption is that the well-established dynamic programming solution methods for influence diagrams cannot be used. For this purpose, Lauritzen and Nilsson (2001) present *single policy update*, a solution heuristic that finds a locally optimal strategy for a LIMID. However, these solutions have no guarantee for optimality unless specific conditions are met. Exact solution methods for finding a maximum expected utility (MEU) solution for a LIMID were first proposed by de Campos and Ji (2012) and Mauá et al. (2012). We will instead next discuss two concomitantly developed mixed-integer formulations for solving limited memory influence diagrams. Compared to the previous approaches, these two have significant advantages, such as being able to consider different constraints and objective functions, including chance constraints and conditional value-at-risk (CVaR), while still guaranteeing the optimality of solutions due to the problems being represented

as MIPs.

2.2.1 Decision Programming

Salo et al. (2022) present Decision Programming, a framework based on a mixed-integer linear reformulation of LIMIDs. Their model is based on the concept of paths $s = (s_j)_{j \in N^C \cup N^D}$ consisting of decision and chance node states, and Paper I presents various improvements to this path-based model. For a given strategy $Z = (Z_j)_{j \in N^D}$, the influence diagram can be seen as a Bayesian network (Koller and Friedman, 2009), and the chain rule

$$\mathbb{P}(X_s) = \prod_{j \in N^C \cup N^D} \mathbb{P}(X_j = s_j \mid X_{I(j)} = s_{I(j)})$$

can be applied to obtain the probability of a path. Importantly, we have defined a conditional probability distribution for all chance and decision nodes, and the probability of a path s given a strategy Z can be calculated as

$$\mathbb{P}(s \mid Z) = \prod_{c \in N^C} \mathbb{P}(X_c = s_c \mid X_{I(c)} = s_{I(c)}) \prod_{d \in N^D} \mathbb{I}(s_{I(d)}, s_d).$$

Recall that we assume the value nodes $v \in N^V$ to be deterministic mappings from each information state $s_{I(v)} \in S_{I(v)}$. For any path s, we can thus obtain the utility $U(s)$ as $\sum_{v \in N^V} U_v(s_{I(v)})$. To explain the formulation, we will consider a MEU problem, but it should be noted that using, e.g., CVaR in the objective function is relatively straightforward, as shown in Salo et al. (2022). Suppose we are interested in maximizing the expected utility $EU(Z) = \sum_{s \in S} \mathbb{P}(s \mid Z) U(s)$. If we introduce binary variables $z(s_j \mid s_{I(j)})$ representing the local decision strategies $Z_j(s_{I(j)})$ such that $z(s_j \mid s_{I(j)}) = 1 \iff \mathbb{I}(s_{I(d)}, s_d) = 1$, a direct approach to the problem would be to solve

$$\max. \sum_{s \in S} p(s) U(s) \prod_{d \in N^D} z(s_d, s_{I(d)}) \tag{2.5}$$

$$\text{s.t.} \sum_{s_d \in S_d} z(s_d \mid s_{I(d)}) = 1, \ \forall d \in N^D, s_{I(d)} \in S_{I(d)} \tag{2.6}$$

$$z(s_d \mid s_{I(d)}) \in \{0, 1\}, \ \forall d \in N^D, s_d \in S_d, s_{I(d)} \in S_{I(d)}, \tag{2.7}$$

where $p(s) = \prod_{c \in N^C} \mathbb{P}(X_c = s_c \mid X_{I(c)} = s_{I(c)})$ for brevity. However, this formulation has a major practical issue: the objective function contains products of $|N^D|$ binary variables, which need to be linearized before applying the efficient solution methods discussed earlier. Salo et al. (2022) tackle this issue by introducing variables $\pi(s)$ and affine constraints forcing

their values to $p(s)\prod_{d\in N^D} z(s_d, s_{I(d)})$. Their formulation then becomes

$$\max. \sum_{s\in S} \pi(s)U(s) \tag{2.8}$$

$$\text{s.t.} \sum_{s_d\in S_d} z(s_d \mid s_{I(d)}) = 1,\ \forall d \in N^D, s_{I(d)} \in S_{I(d)} \tag{2.9}$$

$$0 \leq \pi(s) \leq p(s),\ \forall s \in S \tag{2.10}$$

$$\pi(s) \leq z(s_d \mid s_{I(d)}), \forall s \in S, d \in N^D \tag{2.11}$$

$$\pi(s) \geq p(s) + \sum_{d\in N^D} z(s_d \mid s_{I(d)}) - |N^D|,\ \forall s \in S \tag{2.12}$$

$$z(s_d \mid s_{I(d)}) \in \{0,1\},\ \forall d \in N^D, s_d \in S_d, s_{I(d)} \in S_{I(d)}, \tag{2.13}$$

where constraint (2.11) sets $\pi(s) = 0$ if the path s is not *compatible* with the strategy Z defined by the z-variables. Constraints (2.10) and (2.12) then set $\pi(s) = p(s)$ for paths s that are compatible with Z, that is, paths for which $\sum_{d\in N^D} z(s_d \mid s_{I(d)}) = |N^D|$. This is a mixed-integer linear formulation for which branch-and-cut methods can be used to find the solution, and Salo et al. (2022) show how CVaR or chance constraints can be added to the formulation while still preserving the mixed-integer linear nature of the models.

2.2.2 Rooted junction trees

In their introduction of Decision Programming, Salo et al. (2022) discuss the computational performance of the formulation, and one of the conclusions is that the model (2.8)-(2.13) suffers greatly from the curse of dimensionality. While the path-based approach makes different extensions easy, it is also a major computational roadblock, as the size of the model inevitably becomes exponential in the number of decision and chance nodes. Parmentier et al. (2020) circumvent the exponential growth of the model by first converting the LIMID to an equivalent *gradual rooted junction tree* (G-RJT), then reformulating the junction tree as a mixed-integer problem. We will next present a summary of G-RJTs, and for a thorough description, we refer the reader to Parmentier et al. (2020).

A G-RJT $\mathscr{G} = (\mathscr{V}, \mathscr{A})$ is a directed acyclic graph, much like an influence diagram. Instead of individual chance/decision/value nodes, a junction tree consists of clusters $C \in \mathscr{V}$, each containing a subset of the nodes $j \in N$ from the influence diagram. We first specify that $\mathscr{G}$ must be a rooted tree. For this graph to be a G-RJT, allowing for the upcoming formulation, it must first satisfy the running intersection property. This states that for any two distinct clusters C_1 and C_2 in $\mathscr{V}$, every cluster C on the undirected path between C_1 and C_2 must satisfy $C_1 \cap C_2 \subset C$. Next, we define the *root cluster* of a node $j \in N$ as the root of the subgraph induced by the node j

and further require that each cluster is the root cluster of exactly one node $j \in N$. Finally, for each cluster, we require $I(j) \subset C_j$, where C_j is the root cluster of node $j \in N$. The G-RJT corresponding to the pig farm problem in Fig. 2.2 is presented in Fig. 2.3.

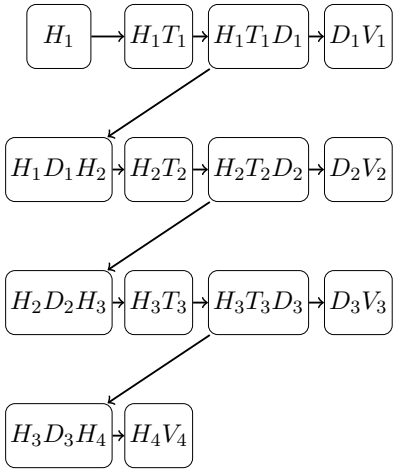


Figure 2.3. The gradual rooted junction tree of the pig farm problem.

We will next present the mixed-integer formulation corresponding to a G-RJT representation of a LIMID. Instead of probability variables over the full set of paths as in Decision Programming, we consider probability distributions μ_{C_j} over each cluster $C_j \in \mathscr{V}$. These properties ensure that we can tie the distributions of adjacent clusters (that is, $(C_1, C_2) \in \mathscr{A}$) together and, starting from the trivial distribution of the root cluster, build the distribution of each cluster. The properties listed above result in the root cluster having only one node. For the pig farm problem in Fig. 2.3, the root node/cluster is H_1, and $\mu_{C_{H_1}}(s_{H_1}) = \mathbb{P}(X_{H_1} = s_{H_1})$. For the next cluster, $\mu_{C_{T_1}}(s_{H_1}, s_{T_1}) = \mathbb{P}(X_{T_1} = s_{T_1} \mid X_{H_1} = s_{H_1})\mu_{C_{H_1}}(s_{H_1})$. Finally, for any pair of adjacent clusters, we require that the marginal distribution over $S_{C_1 \cap C_2}$ is the same whether it is obtained from μ_{C_1} or μ_{C_2} to ensure that the distributions are updated correctly as we move along the directed path(s) from the root cluster. The resulting formulation is

$$\text{max.} \sum_{v \in N^V} \sum_{s_{C_v} \in S_{C_v}} \mu_{C_v}(s_{C_v}) u_{C_v}(s_{C_v}) \tag{2.14}$$

$$\text{s.t.} \sum_{s_{C_j} \in S_{C_j}} \mu_{C_j}(s_{C_j}) = 1, \ \forall j \in N \tag{2.15}$$

$$\sum_{\substack{s_{C_i} \in S_{C_i}, \\ s_{C_i \cap C_j} = s^*_{C_i \cap C_j}}} \mu_{C_i}(s_{C_i}) = \sum_{\substack{s_{C_j} \in S_{C_j}, \\ s_{C_i \cap C_j} = s^*_{C_i \cap C_j}}} \mu_{C_j}(s_{C_j}),\ \forall (C_i, C_j) \in \mathscr{A}, s^*_{C_i \cap C_j} \in S_{C_i \cap C_j} \tag{2.16}$$

$$\mu_{C_j}(s_{C_j}) = \mu_{\overline{C}_j}(s_{\overline{C}_j}) \mathbb{P}(X_j = s_j \mid X_{I(j)} = s_{I(j)}),\ \forall j \in N^C \cup N^V, s_{C_j} \in S_{C_j} \tag{2.17}$$

$$\mu_{C_j}(s_{C_j}) = \mu_{\overline{C}_j}(s_{\overline{C}_j}) z(s_j, s_{I(j)}),\ \forall j \in N^D, s_{C_j} \in S_{C_j} \tag{2.18}$$

$$\mu_{C_j}(s_{C_j}) \geq 0,\ \forall j \in N, s_{C_j} \in S_{C_j} \tag{2.19}$$

$$z(s_j, s_{I(j)}) \in \{0, 1\},\ \forall j \in N^D, s_j \in S_j, s_{I(j)} \in S_{I(j)}, \tag{2.20}$$

where $z(s_j, s_{I(j)})$ represent the decision strategy $Z \in \mathbb{Z}$ as before and $u_{C_v}(s_{C_v})$ is the utility from the cluster corresponding to a value node $v \in N^V$. Constraints (2.18) involve a product of a continuous variable μ and a binary variable z, which technically makes the model nonlinear. However, these constraints can be seen as indicator constraints and reformulated using affine big-M constraints (Parmentier et al., 2020) or other techniques discussed in Bonami et al. (2015).

For large models, constraints (2.16) predominate the total number of constraints in (2.14)-(2.20). From the properties of a G-RJT listed above, it can be seen that for $(C_i, C_j) \in A_{RJT}$, we have $I(j) \subseteq C_i \cap C_j$, and the number of constraints in the model is thus $\mathcal{O}(|S_{I(j)}|)$. Taking this a step further, we define the width of a junction tree as the size of the largest cluster minus one. We then see that because $(I(j) \cup j) \subseteq C_j$ by construction, the rooted treewidth (the minimum width of an RJT on a LIMID) is greater or equal to the size of the largest information set. The number of constraints in (2.14)-(2.20) thus grows exponentially with the width of the tree, not the number of nodes. For problems with a moderate treewidth, this results in vastly superior computational performance compared to Decision Programming. While the approach of considering clusters and their probability distribution makes the formulation more efficient, it has the disadvantage that, unlike with Decision Programming, the full probability/utility distributions are not readily available in the model. This, in turn, prevents directly formulating, e.g., chance constraints and CVaR, further discussed in Paper II.

2.3 Bilevel optimization

The last part of this dissertation, Paper IV, explores the novel field of trilevel optimization. Before we can discuss trilevel, or even bilevel optimization, it is necessary to first discuss the concept of (Lagrangian) duality. We start the discussion with linear problems, mainly because the following

motivation is natural for LP problems. However, the results generalize to convex problems satisfying a constraint qualification. Notably, Dorn (1960) presents similar dual formulations for quadratic problems.

For a linear problem $\max\{c^\top x : Ax \leq b, x \geq 0\}$ we can exploit the information in the constraints to obtain an upper bound for the objective function $c^\top x$. Assuming that $Ax \leq b$ holds, it is evident that a weighted combination $p^\top Ax \leq p^\top b$ of the constraints also holds for any vector $p \geq 0$. Furthermore, because we only consider non-negative variables x, if $c^\top \leq p^\top A$, we have $c^\top x \leq p^\top Ax \leq p^\top b$, and $p^\top b$ is thus an upper bound for $c^\top x$. What remains is to make the bound as tight (small) as possible, and we obtain the Lagrangian dual problem

$$\text{min. } b^\top p \tag{2.21}$$

$$\text{s.t. } A^\top p \geq c \tag{2.22}$$

$$p \geq 0. \tag{2.23}$$

It is easy to see that, by construction, we have $c^\top x \leq b^\top p$, a property known as *weak duality*. For linear problems, we additionally have *strong duality*, that is, the optimal objective values of the primal and dual problems are equal. From the primal and dual problems, we can finally arrive at two results allowing us to replace an optimization problem with a set of constraints for which any feasible solution is optimal to the original problem.

First, we can use strong duality and observe that any solution satisfying primal and dual feasibility with equal primal and dual objective values is primal-dual optimal. For the problem stated above, these conditions become

$$c^\top x = b^\top p \tag{2.24}$$

$$Ax \leq b \tag{2.25}$$

$$A^\top p \geq c \tag{2.26}$$

$$x, p \geq 0. \tag{2.27}$$

Alternatively, we can use *complementary slackness*, which states that for linear problems at the primal-dual optimal solution, any primal/dual constraint corresponding to a nonzero dual/primal variable must be active (that is, satisfied as an equality), and any primal/dual variable corresponding to an inactive dual/primal constraint must be zero. This can be represented using a *complementarity constraint* $0 \leq a \perp b \geq 0$, stating that two non-negative vectors a and b are perpendicular to each other, that is, $a^\top b = 0$. Note that for non-negative vectors, perpendicularity implies that either a_i or b_i must be zero for all $i \in \{1, ..., n\}$, where n is the length of vectors a and b. A primal-dual optimal solution can thus be obtained

from

$$0 \leq x \perp A^\top p - c \geq 0 \tag{2.28}$$

$$0 \leq p \perp b - Ax \geq 0, \tag{2.29}$$

where complementary slackness and primal/dual feasibility are combined in the complementarity constraints. The constraints (2.28) and (2.29) form a linear complementarity problem (LCP), for which specialized solvers such as PATH (Dirkse and Ferris, 1995) exist. These *optimality conditions* state that any solution (x^*, p^*) satisfying either the constraints (2.24)-(2.27) or (2.28)-(2.29) is an optimal solution to the corresponding primal and dual problems. That is, solving the optimization problem and its dual is equivalent to finding a solution satisfying a set of constraints.

To consider more general problems, one can use the results of Karush (1939) and Kuhn and Tucker (1951), referred to as the Karush-Kuhn-Tucker (KKT) conditions. The KKT conditions can be considered an extension of the above discussion, combining primal and dual feasibility with complementary slackness for a general problem of the form

$$\text{min. } f(x) \tag{2.30}$$

$$\text{s.t. } g_i(x) \leq 0, \ \forall i \in I \tag{2.31}$$

$$h_j(x) = 0, \ \forall j \in J \tag{2.32}$$

$$x \geq 0. \tag{2.33}$$

For this problem, the KKT conditions become

$$0 \leq x \perp \nabla f(x) + \sum_{i \in I} \lambda_i \nabla g_i(x) + \sum_{j \in J} \mu_j \nabla h_i(x) \geq 0 \tag{2.34}$$

$$0 \leq \lambda \perp -g(x) \geq 0 \tag{2.35}$$

$$h(x) = 0, \tag{2.36}$$

where for notational compactness, x, g, h, λ and μ are assumed to be vectors. For convex problems satisfying a constraint qualification, the KKT conditions are both necessary and sufficient for optimality (Bertsekas, 2016).

The main practical challenge with KKT conditions is often the complementarity, which tends to result in non-convexity, requiring more sophisticated and, consequently, more computationally demanding solution methods. Multiple approaches exist for modeling the complementarity constraints; notable examples include SOS1 constraints (Beale and Tomlin, 1970; Siddiqui and Gabriel, 2013) and Fortuny-Amat big-M constraints (Fortuny-Amat and McCarl, 1981; Pineda and Morales, 2019). However, the complementarity constraints can be avoided altogether if one instead

constructs the dual of (2.30)-(2.33) and observes that (again, for convex problems satisfying a constraint qualification) strong duality holds and the primal and dual solutions are equal at optimum.

Finally, these results can be used to rationalize the idea of bilevel optimization. In bilevel optimization, as the name suggests, two hierarchical levels of decision making are considered. The *upper-level player* has an objective function and constraints as usual, but must also take into account their impact on the *lower-level player(s)* whose decisions impact the upper-level objective value and/or constraints. A relevant example for this dissertation and bilevel optimization literature in general is energy market modeling, where the lower-level players are profit-maximizing electricity generators and the upper-level player is a transmission system operator aiming to maximize social welfare or profit. The upper-level player then makes decisions on, e.g., transmission capacity expansion, taking into account the effect this will have on the electricity producers' strategies and, consequently, social welfare.

The two main solution approaches for bilevel optimization are based on using either complementarity or strong duality to add the lower-level problem(s) to the upper-level problem as constraints. As discussed earlier, using the KKT conditions introduces complementarity constraints within this single-level reformulation, resulting in a *mathematical programming with equilibrium constraints* (MPEC) problem, and a suitable solver such as KNITRO (Artelys, 2023) or NLPEC (GAMS Development, 2023) can be used to obtain solutions. Enforcing primal and dual feasibility along with strong duality instead results in a *mathematical programming with primal and dual constraints* (MPPDC) formulation of the problem. Because of the lack of complementarity constraints, such models can be solved using more general-purpose mathematical programming solvers.

3. Contributions

In both mixed-integer and bilevel optimization, decision problems can admit multiple alternative representations and the choice of modeling approach can have a major impact on computational performance. For MIPs, as discussed in Section 2.1, if an exact representation of the convex hull of the feasible set was available, the problem could be solved as a single linear programming problem. Extending this idea, if a formulation closely approximates the convex hull, the initial LP relaxation solution is likely to be close to the optimum and not require significant computational effort in terms of branch-and-bound and/or cutting planes before reaching the integer optimal point. Similarly, for bilevel optimization, the choice of using a complementarity or strong duality -based reformulation determines the type of solver required for solving the model, which can have a major impact on the computational performance.

Table 3.1 summarizes the individual contributions of Papers I-IV. These individual papers intersect both in methodology and applications, focusing on developing novel mixed-integer formulations for problems arising from energy-environmental decision making. Papers I and II concentrate on further developing two recently proposed MIP reformulations (Salo et al., 2022; Parmentier et al., 2020) for limited memory influence diagrams (LIMIDs), allowing one to efficiently solve problems with decision-dependent probabilities (also known as type 1 endogenous uncertainty). Notably, Paper I presents an improved MIP reformulation for influence diagrams, enhancing computational performance compared to the model in Salo et al. (2022), while Paper II instead focuses on extending the modeling capabilities of the rooted junction tree reformulation in Parmentier et al. (2020).

Paper III shows how conditionally observed information (type 2 endogenous uncertainty) can be considered in the LIMID reformulations. The combined developments in the first three papers are also applied to a large-scale climate change mitigation case study comprising both types of endogenous uncertainty in Paper III. These two types of endogenous uncertainty have not been considered together in the prior literature, and

Paper	Objectives	Methodologies	Results
I: Solving influence diagrams via efficient mixed-integer programming formulations and heuristics	To explore the computational impact of tighter MIP formulations and heuristics for Decision Programming	Mixed-integer linear programming, primal heuristics, Decision Programming	The proposed novel formulation is an order of magnitude faster and the implemented heuristic produces good solutions in a very short time compared to solving the full model to optimality.
II: Risk-averse decision strategies for influence diagrams using rooted junction trees	To bring the different constraints and objectives from Decision Programming to the RJT models	Mixed-integer programming, rooted junction trees, risk measures	Conditional value-at-risk, chance constraints and other constraints are implemented for an example problem using the RJT formulation.
III: Solving decision problems with endogenous uncertainty and conditional information revelation using influence diagrams	To provide a methodology for including conditional information revelation in RJT models	Mixed-integer programming, rooted junction trees, cost-benefit analysis	Two different methods of incorporating conditional information revelation in RJT-based models. A climate change mitigation case study is solved and the resulting emission strategies are discussed.
IV: A novel strong duality -based reformulation for trilevel infrastructure models in energy systems development	To explore the novel class of trilevel optimization problems, present a novel single-level reformulation and demonstrate an illustrative energy system model	Equilibrium modeling, primal and dual constraints, complementarity constraints	A novel strong duality -based single-level reformulation for trilevel equilibrium problems. The novel formulation outperforms the previously proposed formulation and is demonstrated using a case study representing the Nordic power market.

Table 3.1. Summary of publications

the developments in these papers thus allow for a potentially broad range of new applications.

Finally, Paper IV considers a different class of problems with multiple

decision makers in a hierarchical structure. In our example problem, this structure arises from a more localized system perspective of climate change mitigation. On the top level, an international policy-maker such as the European Union sets a carbon tax incentivizing the bottom-level electricity producers to lower their production emissions. At the middle level, transmission system operators control electricity transmission between countries. The main methodological contribution of Paper IV is a novel single-level reformulation of such trilevel problems, using strong duality where (Gabriel et al., 2022) used complementarity. The computational experiments show that the novel reformulation outperforms the previous method both in terms of model size and solution time.

3.1 Paper I

Decision Programming, as presented in Salo et al. (2022), presents a powerful concept of converting influence diagrams to mixed-integer linear formulations. Influence diagrams are intuitive visual representations of complicated decision problems, and powerful MIP solvers have been developed in the past 20 years. The main contributions of Paper I are related to making Decision Programming easier and faster to use. First, it shows how the existing MIP formulation can be made considerably tighter, resulting in obtaining optimal solutions faster. Second, the single policy update heuristic (Lauritzen and Nilsson, 2001) is implemented for Decision Programming, allowing for obtaining locally optimal solutions quickly. Third, a Julia (Bezanson et al., 2017) package is presented, greatly lowering the barriers to entry for new users. Finally, the novel formulation is applied to a coronary heart disease prevention case study from Hynninen et al. (2019).

3.2 Paper II

Despite the novel MIP formulations in Paper I, Decision Programming still suffers from exponential model growth as the influence diagrams become larger, greatly limiting the applicability of the framework in large problems. Parmentier et al. (2020) present a computationally more efficient approach to solving MIP reformulations of influence diagrams using rooted junction trees (RJT), but their contribution is limited to expected utility maximization with no additional risk constraints. Paper II implements the risk measures and constraints in Salo et al. (2022) for RJT models, significantly improving the modeling capabilities of the method. On the other hand, this can also be seen as developing a solution method for influence diagrams that is computationally more efficient than Decision

Programming while still retaining its modeling capabilities.

3.3 Paper III

Influence diagrams offer a natural avenue for modeling decision-dependent probabilities, something that the field of stochastic programming has struggled with. Hellemo et al. (2018) poses decision-dependent probabilities as Type 1 endogenous uncertainty and discusses a Type 2: decision-dependent information structure. A common example of Type 2 endogenous uncertainty is conditional information revelation, often used in multi-stage stochastic programming (MSSP) models such as the ones in Apap and Grossmann (2017). Paper III introduces two approaches for including conditional information revelation in influence diagrams. The results are presented for RJT models but could be implemented for Decision Programming with minimal modifications. Finally, the paper combines type 2 endogenous uncertainty and the contributions from Papers I and II in a case study of climate change mitigation, involving two-stage research projects for reducing abatement costs and revealing information about underlying climate parameters. The integrated assessment model and the data on CO2 abatement costs are from Ekholm (2018), and the data on technological progress is based on expert elicitation in Baker et al. (2015).

3.4 Paper IV

Paper III gives insights on the cost-benefit optimal emission pathways to mitigate climate change but does not discuss any concrete methods for reaching the proposed emission targets. Bilevel and, more recently, trilevel optimization can be used for modeling energy systems and the interactions between different players with competing interests. Paper IV introduces a novel single-level reformulation for trilevel problems and presents an illustrative case study on how such models could be used to find incentives for reducing emissions. The novel reformulation uses a strong duality representation of the bottom-level optimality conditions, outperforming the LCP solution set reformulation presented in Gabriel et al. (2022). In particular, the model in the paper is focused on finding a carbon tax that decreases emissions without significantly decreasing total production and thus increasing prices. For the Nordic case study, the data is from Belyak et al. (2023), who collected the data from the ENTSO-E platform (Hirth et al., 2018).

4. Discussion

This dissertation focuses on developing novel mixed-integer formulations for challenging problems arising in energy-environmental decision making. Papers I-III discuss different approaches to finding optimal solutions to limited memory influence diagrams, and Paper IV considers a system-level perspective on the climate change mitigation case study in Paper III and uses trilevel equilibrium modeling to provide insight on how the emissions targets could be achieved using a carbon tax.

Since their introduction (Lauritzen and Nilsson, 2001), limited memory influence diagrams (LIMIDs) have received moderate attention. However, considering how influence diagrams are regarded as a user-friendly problem representation (Howard and Matheson, 2005), it is surprising that no user interface seems to exist for formulating a LIMID and obtaining the optimal strategy. In Paper I, we describe a Julia (Bezanson et al., 2017) package implementing the Decision Programming framework originally from Salo et al. (2022).

To further improve the computational efficiency of the developed framework, Paper II explores an alternative mixed-integer formulation for solving LIMIDs, introduced in Parmentier et al. (2020). Their approach is based on first converting the LIMID into a gradual rooted junction tree, thus avoiding the exponential growth of the MIP model that makes larger influence diagrams computationally intractable when using Decision Programming. In Paper II, we show that all the risk-related objective functions and constraints presented for Decision Programming in Salo et al. (2022) can be implemented for rooted junction tree (RJT) models. However, it should be noted that introducing these additional elements to the RJT model can come at the cost of significantly increasing the computational complexity compared to the expected utility maximization problem for a LIMID with no additional constraints.

The concept of decision-dependent or endogenous uncertainty has been discussed in the literature for the past 30 years, and a summary can be found in Hellemo et al. (2018). While influence diagrams are a natural representation of decision-dependent probability distributions, a significant

part of research on endogenous uncertainty focuses on conditional information revelation (e.g., Apap and Grossmann, 2017). To bridge the gap between these two types of uncertainty, Paper III discusses two approaches for incorporating conditional information revelation into our influence diagram models, either by modifying the LIMID or the resulting optimization problem. Additionally, Paper III illustrates the extended RJT models in a climate change mitigation cost-benefit analysis, showing how the contributions of Papers I-III can be used to solve large-scale decision problems under endogenous uncertainty.

Finally, Paper IV approaches an energy system problem as a trilevel equilibrium model. To efficiently solve the model, we present a novel reformulation for trilevel problems, based on strong duality, greatly reducing the size of the model compared to the model in Gabriel et al. (2022) as no complementarity constraints need to be further reformulated. A comparison between the novel reformulation and the previous approach in Gabriel et al. (2022) also shows that the proposed approach solves the problems significantly faster. This reformulation is used in an illustrative example of the Nordic electricity market, exploring the effect of a carbon tax on emissions and electricity prices.

The four papers in this dissertation propose novel MIP formulations for different problems, outperforming existing approaches. The main focus is on the formulations themselves, and we use general-purpose, off-the-shelf MIP solvers to obtain solutions to these problems. However, limiting ourselves to MIP formulations and solving them using off-the-shelf solvers can be somewhat restrictive. First, as discussed earlier, computational efficiency can greatly vary between two MIP formulations of the same problem. This idea can be extended further, as some decision problems formulated as MIP models could also be implemented as, e.g., Boolean satisfiability problems (Manquinho et al., 1998) with different solution methods and computational performance. Second, for limited memory influence diagrams, alternative solution methods have been presented in de Campos and Ji (2012) and Mauá et al. (2012), but it should be noted that they only consider expected utility maximization with no additional constraints and thus would need further research to allow for the same modeling flexibility as Decision Programming or the RJT models.

Additionally, more specialized approaches for solving MIP problems have been developed. For example, the computational issues arising from the large number of constraints in Decision Programming could possibly be mitigated using methods such as Benders decomposition (Laporte and Louveaux, 1993) or Lagrangian decomposition (Guignard and Kim, 1987). Similarly, for the trilevel models, column-and-constraint generation (Dvorkin et al., 2017) and bilevel branch-and-bound (Fischetti et al., 2018) could yield significant performance improvements.

A major benefit of using mathematical programming to represent deci-

sion problems is the ability to incorporate a variety of different modeling paradigms. For LIMID models, we show that risk can be explicitly considered using chance constraints or CVaR, but further research is still needed for, e.g., robust or bilevel optimization over influence diagrams. The previous work on multi-agent influence diagrams (Koller and Milch, 2003) provides a starting point for bilevel influence diagrams, and robust counterparts to linear programming problems are discussed in, e.g., Ben-Tal et al. (2009).

Combining the developments in this dissertation for influence diagrams and hierarchical decision models would allow for better representing decision-dependent uncertainties in bilevel or even trilevel models. A seemingly simple approach would be to consider bilevel problems where the lower level is represented with an influence diagram. However, we only discuss the case of convex quadratic bottom-level problems in Paper IV, and considering MIP models instead would require accommodating algorithms similar to those in, e.g., Fischetti et al. (2017) and Fischetti et al. (2018).

Finally, the case studies presented in these papers should be viewed as illustrative, and their results are not meant to be directly used to support decision-making. Nevertheless, we find that the results of our example problems are in line with the related literature (Ekholm and Baker, 2022; Hynninen et al., 2019; Belyak et al., 2023), suggesting that the models can be used to obtain sensible results for difficult decision problems. Making the case studies more detailed and realistic would be a promising research avenue yielding interesting policy results.

References

Robert M Apap and Ignacio E Grossmann. Models and computational strategies for multistage stochastic programming under endogenous and exogenous uncertainties. *Computers & Chemical Engineering*, 103:233–274, 2017.

Artelys. KNITRO reference manual, 2023. URL https://www.artelys.com/app/docs/knitro/3_referenceManual.html.

Erin Baker, Valentina Bosetti, Laura Diaz Anadon, Max Henrion, and Lara Aleluia Reis. Future costs of key low-carbon energy technologies: Harmonization and aggregation of energy technology expert elicitation data. *Energy Policy*, 80:219–232, 2015.

Evelyn Martin Lansdowne Beale and John A Tomlin. Special facilities in a general mathematical programming system for non-convex problems using ordered sets of variables. In *Proceedings of the Fifth International Conference on Operational Research*, pages 447–454, 1970.

Richard Bellman. The theory of dynamic programming. *Bulletin of the American Mathematical Society*, 60(6):503–515, 1954.

Nikita Belyak, Steven A. Gabriel, Nikolay Khabarov, and Fabricio Oliveira. Optimal transmission expansion planning in the context of renewable energy integration policies, 2023. URL https://arxiv.org/abs/2302.10562.

Aharon Ben-Tal, Laurent El Ghaoui, and Arkadi Nemirovski. *Robust optimization*. Princeton university press, 2009.

Timo Berthold. *Primal heuristics for mixed integer programs*. Master's thesis, Zuse Institute Berlin (ZIB), 2006.

Dimitri P. Bertsekas. *Nonlinear Programming*. Athena Scientific, 2016.

Jeff Bezanson, Alan Edelman, Stefan Karpinski, and Viral B Shah. Julia: A fresh approach to numerical computing. *SIAM Review*, 59(1):65–98, 2017.

John R Birge. The value of the stochastic solution in stochastic linear programs with fixed recourse. *Mathematical programming*, 24:314–325, 1982.

Pierre Bonami, Andrea Lodi, Andrea Tramontani, and Sven Wiese. On mathematical programming with indicator constraints. *Mathematical programming*, 151:191–223, 2015.

Cplex, IBM ILOG. User's Manual for CPLEX, 2022. URL https://www.ibm.com/docs/en/icos/22.1.1?topic=optimizers-users-manual-cplex.

George B Dantzig. Linear programming under uncertainty. *Management science*, 1(3-4):197–206, 1955.

George B Dantzig, Alex Orden, and Philip Wolfe. Notes on linear programming: Part 1. the generalized simplex method for minimizing a linear form under linear inequality restraints. Technical report, Tech. rep., RAND Corp Santa Monica CA, 1954.

Cassio Polpo de Campos and Qiang Ji. Strategy selection in influence diagrams using imprecise probabilities. *arXiv preprint arXiv:1206.3246*, 2012.

Steven P Dirkse and Michael C Ferris. The PATH solver: a nonmonotone stabilization scheme for mixed complementarity problems. *Optimization methods and software*, 5(2):123–156, 1995.

William S Dorn. Duality in quadratic programming. *Quarterly of applied mathematics*, 18(2):155–162, 1960.

Aryeh Dvoretzky, J Kiefer, and Jacob Wolfowitz. The inventory problem: II. Case of unknown distributions of demand. *Econometrica: Journal of the Econometric Society*, pages 450–466, 1952.

Yury Dvorkin, Ricardo Fernandez-Blanco, Yishen Wang, Bolun Xu, Daniel S Kirschen, Hrvoje Pandžić, Jean-Paul Watson, and Cesar A Silva-Monroy. Co-planning of investments in transmission and merchant energy storage. *IEEE Transactions on Power Systems*, 33(1):245–256, 2017.

Tommi Ekholm. Hedging the climate sensitivity risks of a temperature target. *Climatic Change*, 127(2):153–167, 2014.

Tommi Ekholm. Climatic Cost-benefit Analysis Under Uncertainty and Learning on Climate Sensitivity and Damages. *Ecological Economics*, 154:99–106, 2018.

Tommi Ekholm and Erin Baker. Multiple Beliefs, Dominance and Dynamic Consistency. *Management Science*, 68(1):529–540, 2022.

Fair Isaac Corporation. FICO Xpress Optimizer reference manual, 2024. URL `https://www.fico.com/fico-xpress-optimization/docs/latest/solver/optimizer/HTML/GUID-3BEAAE64-B07F-302C-B880-A11C2C4AF4F6.html`.

Matteo Fischetti, Ivana Ljubić, Michele Monaci, and Markus Sinnl. A new general-purpose algorithm for mixed-integer bilevel linear programs. *Operations Research*, 65(6):1615–1637, 2017.

Matteo Fischetti, Ivana Ljubić, Michele Monaci, and Markus Sinnl. On the use of intersection cuts for bilevel optimization. *Mathematical Programming*, 172 (1-2):77–103, 2018.

José Fortuny-Amat and Bruce McCarl. A representation and economic interpretation of a two-level programming problem. *Journal of the operational Research Society*, 32:783–792, 1981.

Steven A. Gabriel, Marina Leal, and Martin Schmidt. On linear bilevel optimization problems with complementarity-constrained lower levels. *Journal of the Operational Research Society*, 73(12):2706–2716, 2022.

GAMS Development. GAMS/NLPEC reference manual, 2023. URL `https://www.gams.com/latest/docs/S_NLPEC.html`.

Ralph E Gomory. Outline of an algorithm for integer solutions to linear programs and an algorithm for the mixed integer problem. *Bulletin of the American Mathematical Society*, 64:275–278, 1958.

Monique Guignard and Siwhan Kim. Lagrangean decomposition: A model yielding stronger lagrangean bounds. *Mathematical programming*, 39(2):215–228, 1987.

Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2023. URL https://www.gurobi.com.

Lars Hellemo, Paul I Barton, and Asgeir Tomasgard. Decision-dependent probabilities in stochastic programs with recourse. *Computational Management Science*, 15:369–395, 2018.

Lion Hirth, Jonathan Mühlenpfordt, and Marisa Bulkeley. The ENTSO-E Transparency Platform–A review of Europe's most ambitious electricity data platform. *Applied energy*, 225:1054–1067, 2018.

Ronald A Howard and James E Matheson. Influence diagrams. *Decision Analysis*, 2(3):127–143, 2005.

Yrjänä Hynninen, Miika Linna, and Eeva Vilkkumaa. Value of genetic testing in the prevention of coronary heart disease events. *PloS one*, 14(1):e0210010, 2019.

William Karush. Minima of functions of several variables with inequalities as side conditions. Master's thesis, University of Chicago, 12 1939.

Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

Daphne Koller and Brian Milch. Multi-agent influence diagrams for representing and solving games. *Games and economic behavior*, 45(1):181–221, 2003.

Angela Köppl and Margit Schratzenstaller. Carbon taxation: A review of the empirical literature. *Journal of Economic Surveys*, 37(4):1353–1388, 2023.

Harold W. Kuhn and Albert W. Tucker. Nonlinear programming. *Berkeley Symposium on Mathematical Statistics and Probability*, 1951:481–492, 1951.

Ailsa H Land and Alison G Doig. An automatic method for solving discrete programming problems. *Econometrica, Journal of the Econometric Society*, pages 497–520, 1960.

Gilbert Laporte and François V Louveaux. The integer l-shaped method for stochastic integer programs with complete recourse. *Operations research letters*, 13(3):133–142, 1993.

Steffen L Lauritzen and Dennis Nilsson. Representing and solving decision problems with limited information. *Management Science*, 47(9):1235–1251, 2001.

Jon Lee and Sven Leyffer. *Mixed integer nonlinear programming*, volume 154. Springer Science & Business Media, 2011.

Hoesung Lee et al. IPCC, 2023: Climate Change 2023: Synthesis Report, Summary for Policymakers. Contribution of Working Groups I, II and III to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change [Core Writing Team, H. Lee and J. Romero (eds.)]. IPCC, Geneva, Switzerland. Technical Report 10.59327/IPCC/AR6-9789291691647.001, Intergovernmental Panel on Climate Change (IPCC), Geneva, Switzerland, 2023. URL https://mural.maynoothuniversity.ie/17886/. Please access report for extended list of writing team, editors, scientific steering committee and other contributing authors.

Boqiang Lin and Xuehui Li. The effect of carbon tax on per capita CO2 emissions. *Energy policy*, 39(9):5137–5146, 2011.

László Lovász and Alexander Schrijver. Cones of matrices and set-functions and 0–1 optimization. *SIAM journal on optimization*, 1(2):166–190, 1991.

Vasco M Manquinho, João P Marques Silva, Arlindo L Oliveira, and Karem A Sakallah. Satisfiability-based algorithms for 0-1 integer programming. In *Proceedings of the IEEE/ACM International Workshop on Logic Synthesis (IWLS)*, 1998.

Denis Deratani Mauá, Cassio Polpo de Campos, and Marco Zaffalon. Solving limited memory influence diagrams. *Journal of Artificial Intelligence Research*, 44:97–140, 2012.

Sanjay Mehrotra. On the implementation of a primal-dual interior point method. *SIAM Journal on optimization*, 2(4):575–601, 1992.

William D Nordhaus. To Slow or Not to Slow: The Economics of The Greenhouse Effect. *The Economic Journal*, 101(407):920, 1991.

Manfred Padberg and Giovanni Rinaldi. A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems. *SIAM review*, 33(1):60–100, 1991.

Axel Parmentier, Victor Cohen, Vincent Leclère, Guillaume Obozinski, and Joseph Salmon. Integer programming on the junction tree polytope for influence diagrams. *INFORMS Journal on Optimization*, 2(3):209–228, 2020.

Russel J Petch and Said Salhi. A multi-phase constructive heuristic for the vehicle routing problem with multiple trips. *Discrete Applied Mathematics*, 133(1-3): 69–92, 2003.

Salvador Pineda and Juan Miguel Morales. Solving linear bilevel problems using big-Ms: Not all that glitters is gold. *IEEE Transactions on Power Systems*, 34 (3):2469–2471, 2019.

Ramesh Raman and Ignacio E Grossmann. Modelling and computational techniques for logic based integer programming. *Computers & Chemical Engineering*, 18(7):563–578, 1994.

Ahti Salo, Juho Andelmin, and Fabricio Oliveira. Decision programming for mixed-integer multi-stage optimization under uncertainty. *European Journal of Operational Research*, 299(2):550–565, 2022.

Sauleh Siddiqui and Steven A Gabriel. An SOS1-based approach for solving MPECs with a natural gas market application. *Networks and Spatial Economics*, 13:205–227, 2013.

Sun Tzu. *The Art of War*. Watkins Publishing, 2005. ISBN 9781780282992.

United Nations Framework Convention on Climate Change UNFCCC. Paris agreement, 2015. Available at https://unfccc.int/process-and-meetings/the-paris-agreement [Accessed 15.1.2024].

Alex Van Breedam. Improvement heuristics for the vehicle routing problem based on simulated annealing. *European Journal of Operational Research*, 86(3): 480–490, 1995.

John Von Neumann and Oskar Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, 1947.

Abraham Wald. Foundations of a general theory of sequential decision functions. *Econometrica, Journal of the Econometric Society*, pages 279–313, 1947.

Laurence A Wolsey. *Integer programming*. John Wiley & Sons, 2020.

World Climate Conference. Declaration of the World Climate Conference. *Environmental Conservation*, 6(2):137–138, aug 1979. ISSN 0376-8929. URL https://www.cambridge.org/core/product/identifier/S0376892900002630/type/journal{_}article.

Publication I

Helmi Hankimaa, Olli Herrala, Fabricio Oliveira and Jaan Tollander de Balsch. Solving influence diagrams via efficient mixed-integer programming formulations and heuristics. Submitted to *INFORMS Journal on Computing*, December 2023.

Solving influence diagrams via efficient mixed-integer programming formulations and heuristics

Helmi Hankimaa, Olli Herrala, Fabricio Oliveira, Jaan Tollander de Balsch
Department of Mathematics and Systems Analysis, Aalto University, Espoo, Finland

Abstract

In this paper, we propose novel mixed-integer linear programming (MIP) formulations to model decision problems posed as influence diagrams. We also present a novel heuristic that can be employed to warm start the MIP solver, as well as provide heuristic solutions to more computationally challenging problems. All of those improvements are implemented in `DecisionProgramming.jl`, a new Julia package for modelling decision problems as MIP equivalents using the proposed formulations. We provide computational results showcasing the superior performance of these improved formulations as well as the performance of the proposed heuristic. Lastly, we describe a novel case study showcasing decision programming as an alternative framework for modelling multi-stage stochastic dynamic programming problems.

Keywords: decision problems under uncertainty, influence diagrams, decision analysis, mixed-integer programming

1. Introduction

A powerful way to tackle complex real-life decision-making problems is to frame them as multi-stage decision problems under uncertainty. This helps one to understand the interactions between parts of the decision process and how uncertainty behaves from a rigorous, quantitative standpoint. State-of-the-art approaches for modeling and solving multi-stage decision problems stem from two main areas: decision analysis and stochastic programming. Each field has provided time-tested modeling and solution approaches with varying degrees of success depending on how uncertainty is modeled.

These modeling approaches require specifying probability values and measurable consequences to uncertain events such that the decisions maximizing an expected utility function value can be made. However, it remains extremely challenging to analytically

model decision problems due to the particularly interdependent nature that decision problems can exhibit. Outcomes of random events change the optimal decisions and decisions can influence the probability distributions of random events. The task of representing such dependencies may be equally challenging, if not more so, than finding the best decisions.

Influence diagrams [7] provide both a formal description of a decision problem and serve as a communication tool which requires minimal technical proficiency. Furthermore, they are useful in conveying structural relationships of the problem in a simple manner and thus crucially bridge the gap between quantitative specifications and qualitative descriptions.

Due to their generality, influence diagrams pervade many modeling-based approaches that require a formal description of relationships between uncertainty, decisions and consequences. Figure 1 shows an influence diagram for the N-monitoring problem, where N agents (A_1 to A_N) must decide whether to countervail an unknown load (L) based on imprecise readings of this load from their respective sensors (R_1 to R_N). The chance of failure (F) is influenced by the unknown load and the eventual decision to countervail the load. The final utility (T) is calculated considering whether the agents intervened and if a failure was observed. As such, this setting represents independent agents who must make decisions based only on observations of the state of the world but without being able to completely know it or share information among themselves.

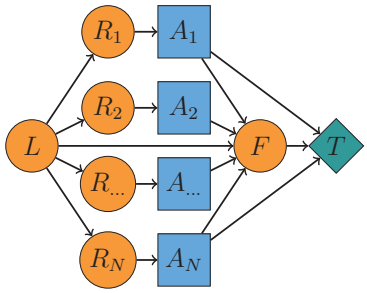


Figure 1: An influence diagram representing the N-monitoring problem. Decisions are represented by squares, chance events by circles and consequences by diamonds

Differently from decision trees, the nodes in an influence diagram do not need to be totally ordered nor do they have to depend directly on all predecessors. This freedom

from dependence on all predecessors allows for the decisions to be made by, e.g., decision-makers who partially observe a common state of information (node L in Figure 1) but may differ in their ability to observe or are incapable of sharing information.

Unfortunately, quantitative methods employed to obtain optimal decisions from influence diagram representations typically require that some of that generality is curbed. Influence diagrams are, in essence, representations of (possibly partially observable) Markov decision processes [13]. Thus, if (i) a single decision-maker is assumed (implying a total ordering among decision nodes) and (ii) the *no-forgetting* assumption holds (implying that each decision node and its direct predecessors influence all successor decision nodes), then Markovian assumptions hold. This, in turn, enables one to solve influence diagrams with well-established techniques, for instance, by forming the equivalent decision tree that can be solved through dynamic programming; or by removing decision and chance nodes from the diagram one by one [4].

As one may suspect, many problems, including that illustrated in Figure 1, violate assumptions (i) and (ii). Indeed, there may be no memory or communication between deciding agents (meaning that they cannot know each other's decisions) or constraints imposed across the diagram, such as budget limitations or logical conditions (e.g., stating that a given action can only occur if a prerequisite project has been started/completed in the past). All of these either violate the assumption that the previous state is "remembered" at a later stage or that all information influencing the decision alternatives is known when making said decision. These limitations create severe deficiencies in representing real-world problems.

Lauritzen and Nilsson [9] proposed an analytical framework to characterise these *limited memory influence diagrams*. Note that the notion of limited memory can also be used to encompass settings with multiple decision-makers, the limited or absent sharing of information being its defining feature (regardless of whether it is due to lack of memory or communication between decision-makers). In any case, limited-memory influence diagrams are essentially influence diagrams that do not satisfy assumptions (i) and (ii). However, the fact that they do not satisfy assumptions (i) and (ii) means that much more sophisticated analysis is required for designing methods that can extract optimal decisions from these diagrams. These methods involve specialised structures such as junction trees and message-passing mechanisms.

Existing approaches for solving influence diagrams are deficient in many ways. First, solution methods are typically based on ad-hoc algorithm implementations that require the user to be proficient in manipulating influence diagrams. Additionally, such methods are often designed only for problems where expected utility is maximized, with no constraints connecting different decisions (e.g., budget/chance constraints). This limits their capabilities in modeling risk-averse decision making. Finally, tackling the challenges related to implementing computationally efficient methods (e.g., memory allocation, and thread-safe parallelisation) is required for larger problems. This creates a significant entry barrier to the wider adoption of such approaches.

It is precisely against this backdrop that the development of *decision programming* [14] started. Decision programming leverages the capabilities of stochastic programming [5] and decision analysis to model and solve multi-stage decision problems using mathematical optimization techniques. In essence, decision programming exploits the expressiveness of influence diagrams in structuring problems to develop deterministic equivalent [5] mixed-integer programming (MIP) formulations. In turn, commercial-grade, professionally developed, off-the-shelf software can be readily utilised for solving these problems in a relatively straightforward manner instead of relying on ad-hoc implementations. It is worth highlighting that the latter, in stark contrast with the former, tend to be problem-specific and provide few guarantees for complying with sound software engineering practices in terms of versioning, updating and continuous improvements.

Concomitantly, the fact that decision programming models can be formulated as MIP problems gives rise to major benefits from the modeling perspective. In particular, decision programming exploits the exceptional modeling expressiveness provided by MIP to tackle challenging decision problems originally posed in the field of decision analysis. Having an underpinning MIP-based approach is extremely timely, especially due to the recent remarkable progress in MIP technology, with some estimating that the combined hardware and software speed-ups amount to a factor of two trillion between the early 1990s and mid-2010s [2]! Thus, there is convincing evidence that MIP approaches for decision analysis problems are practically relevant, perhaps contrary to a few decades ago.

In this paper, we provide multiple contributions in terms of practical and methodological aspects associated with decision programming. First and foremost, we present novel

contributions that further improve upon the original ideas in Salo et al. [14]. Specifically, we present a novel formulation for decision programming problems that is considerably more efficient from a computational standpoint. This is illustrated in the computational experiments we provide. Furthermore, we propose a heuristic inspired by the single policy update heuristic, originally proposed in Lauritzen and Nilsson [9]. This heuristic can be used not only to generate feasible solutions in case of more computationally challenging problems but also can be employed to warm start the MIP model.

We also present `DecisionProgramming.jl`, a Julia [3] package that provides a seamless interface for users to pose decision problems as influence diagrams. The problem is then automatically converted into a MIP formulation using `JuMP.jl` [6], a Julia package for formulating mathematical programming models which provides an interface to a wide range of open-source and commercial mathematical optimization solvers.

Finally, we present a comprehensive case study in which we use the decision programming framework to develop an optimal decision strategy for allocating preventive care for coronary heart disease (CHD). The aim of this study is to evaluate the suitability of decision programming for performing the cost-benefit analysis originally performed by Hynninen et al. [8]. In Hynninen et al. [8], a set of alternative predefined testing and treatment strategies for CHD are optimized using dynamic programming. We show how the same problem can be solved precluding the need to define strategies a priori. This is because all of the possible strategies are within the feasible solution set of the model and, thus, once solved, the solution defines the optimal strategy. This showcases both the benefits of posing such decision problems as MIP formulations and the range of applications that can be tackled by decision programming.

This paper is structured as follows. Section 2 presents the technical details associated with the decision programming framework. In Section 3, we present the novel formulation proposed in this paper, followed by the proposed adaptation of the single policy update heuristic presented in Section 4. In Section 5, we present details of the user interface made available by `DecisionProgramming.jl` and how it allows users to implement decision programming problems based on their influence diagrams. In Section 6, we provide computational results showcasing the benefits of the methodological innovations proposed in this paper and in Section 7 we describe the case study considering optimal preventive care strategies for CHD. Lastly, in Section 8 we provide conclusions and discuss

some potential directions for further research.

2. Decision Programming

Decision programming relies on influence diagrams, which are graphical representations of decision problems. In influence diagrams, nodes represent chance events, decisions and consequences. Specifically, let $G(N, A)$ be an acyclic graph formed by nodes in $N = C \cup D \cup V$, where C is a subset of chance nodes, D a subset of decision nodes, and V a subset of value nodes. Value nodes represent consequences incurred from decisions made at nodes D and chance events observed at nodes C. Each decision and chance node $j \in C \cup D$ can assume a state s_j from a discrete and finite set of states S_j. For a decision node $j \in D$, S_j represents the decision alternatives. For a chance node $j \in C$, S_j is the set of possible outcomes.

In the diagram, arcs represent interdependency among decisions and chance events. Set $A = \{(i, j) \mid i, j \in N\}$ contains the arcs (i, j), which represent the influence between nodes i and j. This influence is propagated in the diagram in the form of *information*. That is, an arc (i, j) that points to a decision node $j \in D$ indicates that the decision at $j \in D$ is made *knowing* the realisation (i.e., uncertainty outcome or decision made) of state $s_i \in S_i$, with $i \in C \cup D$. On the other hand, an arc that points to a chance node $j \in C$ indicates that the realisation $s_j \in S_j$ is dependent (or conditional) on realisation $s_i \in S_i$ of node $i \in C \cup D$.

The *information set* $I(j) = \{i \in N \mid (i, j) \in A\}$ comprises all immediate predecessors (or parents) of a given node $j \in N$. Despite being a less common terminology, we opt for the term "information set" to highlight the role of information in the modeling of the decision process. The decisions $s_j \in S_j$ made in each decision node $j \in D$ depend on their *information state* $s_{I(j)} \in S_{I(j)}$, where $S_{I(j)} = \prod_{i \in I(j)} S_i$ is the set of all possible information states for node j. Analogously, the possible realisations $s_j \in S_j$ for each chance node $j \in C$ and their associated probabilities also depend on their information state $s_{I(j)} \in S_{I(j)}$.

Let us define $X_j \in S_j$ as the realised state at a chance node $j \in C$. For a decision node $j \in D$, let $Z_j : S_{I(j)} \to S_j$ be a mapping between each information state $s_{I(j)} \in S_{I(j)}$ and decision $s_j \in S_j$. That is, $Z_j(s_{I(j)})$ defines a local decision strategy, which represents the choice of some $s_j \in S_j$ in $j \in D$, given the information $s_{I(j)}$. Such a mapping can be

represented by an indicator function $\mathbb{I} : S_{I(j)} \times S_j \to \{0, 1\}$ defined so that

$$\mathbb{I}(s_{I(j)}, s_j) = \begin{cases} 1, & \text{if } Z_j \text{ maps } s_{I(j)} \text{ to } s_j, \text{ i.e., } Z_j(s_{I(j)}) = s_j; \\ 0, & \text{otherwise.} \end{cases}$$

A (global) *decision strategy* is the collection of local decision strategies in all decision nodes: $Z = (Z_j)_{j \in D}$, selected from the set of all possible strategies $\mathbb{Z}$.

A *path* is a sequence of states $s = (s_i)_{i=1,\ldots,n}$, with $n = |C| + |D|$ and

$$S = \{(s_i)_{i=1,\ldots,n} \mid s_i \in S_i, i = 1, \ldots, n\} \tag{1}$$

is the set of all possible paths. We assume that the nodes $C \cup D$ are numbered from 1 to n such that for each arc $(i, j) \in A$, $i < j$. Moreover, we say that a strategy Z is compatible with a path $s \in S$ if $Z_j(s_{I(j)}) = s_j$ for all $j \in D$. We denote as $S(Z) \subseteq S$ the subset of all paths that are compatible with a strategy Z.

Using the notion of information states, the conditional probability of observing a given state s_j for $j \in C$ is $\mathbb{P}(X_j = s_j \mid X_{I(j)} = s_{I(j)})$. The probability associated with a path $s \in S$ being observed given a strategy Z can then be expressed as

$$\mathbb{P}(s \mid Z) = \left(\prod_{j \in C} \mathbb{P}(X_j = s_j \mid X_{I(j)} = s_{I(j)}) \right) \left(\prod_{j \in D} \mathbb{I}(s_{I(j)}, s_j) \right) \tag{2}$$

Notice that the term $\prod_{j \in D} \mathbb{I}(s_{I(j)}, s_j)$ in equation (2) takes value one if the strategy Z is compatible with the path $s \in S$, being zero otherwise. Furthermore, notice that one can pre-calculate the probability

$$p(s) = \left(\prod_{j \in C} \mathbb{P}(X_j = s_j \mid X_{I(j)} = s_{I(j)}) \right). \tag{3}$$

of a path $s \in S$ being observed, in case a compatible strategy is chosen.

At the value node $v \in V$, a real-valued utility function $U_v : S_{I(v)} \to \mathbb{R}$ maps the information state $s_{I(v)}$ to a utility value $U_v(s_{I(v)})$. We usually assume the utility value of a path s to be the sum of individual value nodes' utilities: $U(s) = \sum_{v \in V} U_v(s_{I(v)})$. The default objective is to choose a strategy $Z \in \mathbb{Z}$ maximizing the expected utility, which can be expressed as

$$\max_{Z \in \mathbb{Z}} \sum_{s \in S} \mathbb{P}(s \mid Z) U(s). \tag{4}$$

Notice that other objective functions can also be modeled. For example, Salo et al. [14] discuss the use of the conditional value-at-risk.

To formulate this into a mathematical optimization problem, we start by representing the local strategies Z_j using binary variables $z(s_j \mid s_{I(j)})$ that take value one if $\mathbb{I}(s_{I(j)}, s_j) = 1$, and 0 otherwise. We then observe that using (2) and (3), the objective function (4) becomes

$$\max_z \sum_{s \in S} p(s) U(s) \prod_{j \in D} z(s_j \mid s_{I(j)}).$$

This function is nonlinear, and is used only for illustrating the nature of the formulations. The usefulness of this construction becomes more obvious in Section 3. Salo et al. [14] instead replace the conditional path probability $\mathbb{P}(s \mid Z)$ in (4) with a continuous decision variable $\pi(s)$, enforcing the correct behavior of this variable using affine constraints.

With these building blocks, the problem can be formulated as a mixed-integer linear programming (MILP) model, which allows for employing off-the-shelf mathematical programming solvers. The MILP problem presented in Salo et al. [14] can be stated as (5)-(10).

$$\max_{Z \in \mathbb{Z}} \sum_{s \in S} \pi(s) U(s) \tag{5}$$

$$\text{subject to} \sum_{s_j \in S_j} z(s_j \mid s_{I(j)}) = 1, \qquad \forall j \in D, s_{I(j)} \in S_{I(j)} \tag{6}$$

$$0 \leq \pi(s) \leq p(s), \qquad \forall s \in S \tag{7}$$

$$\pi(s) \leq z(s_j \mid s_{I(j)}), \qquad \forall j \in D, s \in S \tag{8}$$

$$\pi(s) \geq p(s) + \sum_{j \in D} z(s_j \mid s_{I(j)}) - |D|, \qquad \forall s \in S \tag{9}$$

$$z(s_j \mid s_{I(j)}) \in \{0, 1\}, \qquad \forall j \in D, s_j \in S_j, s_{I(j)} \in S_{I(j)}. \tag{10}$$

Variables $\pi(s)$ are nonnegative continuous variables representing the conditional path probability in equation (2). They take the value of the path probability $p(s)$ in case the selected strategy Z is compatible with the path $s \in S$ and zero otherwise. Notice that this compatibility is equivalent to observing $z(s_j \mid s_{I(j)}) = 1$ for all $s_j \in S$ such that $j \in D$.

The objective function (5) defines the expected utility value, which is calculated considering only the paths that are compatible with the strategy. Constraint (6) enforces the one-to-one nature of the mapping $\mathbb{I}(s_{I(j)}, s_j)$, represented by the z-variables. The correct behaviour of variables $\pi(s)$ is guaranteed by constraints (7)-(9), which enforce that $\pi(s) = p(s)$ if $z(s_j \mid s_{I(j)}) = 1$ for all $s_j \in S$ such that $j \in D$. The term $|D|$ in

(9) represents the cardinality of the set D, that is, the number of decision nodes in the diagram. Notice that the domain of $\pi(s)$ is defined in (7).

3. Improved formulations

One key challenge associated with formulation (5)–(10), and, in fact, any MILP formulation, is that computational performance is strongly tied to the tightness of the formulation. In this context, the tightness of a MILP formulation is related to how close the linear relaxation solution is to the initial primal bound, e.g., the first integer feasible solution value obtained by the solver during the solution process or one obtained using primal heuristics.

Next, we present reformulations developed to enhance the numerical performance of the decision programming formulation (5)–(10). For that, let us first define the subset of paths

$$S_{s_j|s_{I(j)}} = \left\{ s \in S \mid (s_{I(j)}, s_j) \subseteq s \right\}.$$

Notice that we use the notation $(s_{I(j)}, s_j)$ to represent a portion of a path s, formed by the combination of the information state $s_{I(j)}$ (which may itself be a collection of states, if $|I(j)| > 1$) and the state s_j. We also utilise the set operator $\subseteq$ to indicate that the states $(s_{I(j)}, s_j)$ are part of the path $s \in S$. Notice that the states $(s_{I(j)}, s_j)$ do not need to be consecutive in the path s, although the ordering between $s_{I(j)}$ and s_j is naturally preserved in s.

Considering $j \in D$, the subset $S_{s_j|s_{I(j)}}$ allows us to define the notion of *locally compatible paths*, that is, the collection of paths s compatible with local strategies Z_j for which $\mathbb{I}(s_{I(j)}, s_j) = 1$. The definition of the subset $S_{s_j|s_{I(j)}}$ allows us to derive the following valid inequality for (5)–(10).

$$\sum_{s \in S_{s_j|s_{I(j)}}} \pi(s) \leq z(s_j \mid s_{I(j)}), \quad \forall j \in D, s_j \in S_j, s_{I(j)} \in S_{I(j)}. \tag{11}$$

Constraint (11) states that only paths that are compatible with the selected strategy might be allowed to have a probability different than zero. Moreover, since it is enforced on all decision nodes, it means that this constraint guarantees that only the paths that are compatible with the strategy Z are active. Recall that we denote this set of compatible paths as $S(Z) \subseteq S$.

As pointed out in Salo et al. [14], for expected utility maximization, constraint (9), which prevents variables $\pi(s)$ from wrongly taking value zero, is only required when some of the utility values $U(s)$, $s \in S$, are negative. Notice that this is otherwise prevented by the maximization of the objective function (5), naturally steering these variables to their upper bound values. Another way to guarantee that the variables $\pi(s)$ take their correct value, i.e., $\pi(s) = p(s)$, if $s \in S(Z)$, is to impose the constraint

$$\sum_{s \in S} \pi(s) = 1. \tag{12}$$

As it will be discussed in Section 6, replacing (8) and (9) with (11) and (12) provides considerable gains in terms of linear relaxation strengthening. Furthermore, we observe that the computational performance can be even further improved by employing a simple variable substitution. Recall that in the original formulation (5)-(10), variables $\pi(s)$ represent the conditional path probability $\mathbb{P}(s \mid Z) = p(s) \prod_{j \in D} z(s_j \mid s_{I(j)})$. If we let $x(s) \in [0, 1]$, $s \in S$ represent the product $\prod_{j \in D} z(s_j \mid s_{I(j)})$, then we can reformulate the problem by substituting $\pi(s) = p(s)x(s)$ for all $s \in S$.

Although $x(s)$, $s \in S$, is continuous, it behaves as a binary variable which takes value one whenever the path is compatible with the strategy and zero, otherwise. This is analogous to the behaviour of variable $\pi(s) \in [0, p(s)]$ in (5)–(10). We highlight that, from a theoretical standpoint, there is no obvious reason for performing such a substitution. On the other hand, we will show that it yields significant practical benefits in terms of computational performance.

Using these x-variables, we can reformulate (11) as

$$\sum_{s \in S_{s_j | s_{I(j)}}} x(s) \leq |S_{s_j | s_{I(j)}}| z(s_j \mid s_{I(j)}), \quad \forall j \in D, s_j \in S_j, s_{I(j)} \in S_{I(j)}, \tag{13}$$

a consequence of $x(s) \in [0, 1]$ and the fact that $z(s_j \mid s_{I(j)})$ must be equal to 1 for $x(s)$ to be positive for $s \in S_{s_j | s_{I(j)}}$.

Constraint (13) can be strengthened further. We note that a path must be in the set of compatible paths $S(Z)$ in order for $x(s)$ to be positive with strategy Z. Using this information, we can infer a tighter upper bound for the number of paths that can be active ($x(s) > 0$) from the set of locally compatible paths. We observe that in a set of compatible paths $S(Z)$, each information state $s_{I(j)}$ maps to exactly one decision alternative s_j for each decision node $j \in D$, in accordance with constraint (6). However,

the set of locally compatible paths for a given pair of information state and decision node state $(s_{I(j)}, s_j)$ of decision node $j \in D$, includes paths for all combinations $(s_{I(k)}, s_k)$ of information states and decisions for the other decision nodes $k \in D \setminus \{j\}$. Hence, only a fraction of the locally compatible paths can be active. The fraction is linked to the number of states $|S_k|$ of the other decision nodes $k \in D \setminus \{j\}$. The number of locally compatible paths that will also be active, i.e., $|S_{s_j|s_{I(j)}} \cap S(Z)|$ can be defined as

$$|S_{s_j|s_{I(j)}} \cap S(Z)| = \frac{|S_{s_j|s_I(j)}|}{\Pi_{k \in D \setminus (\{j\} \cup I(j))} |S_k|}. \tag{14}$$

Notice that the calculation of the number of active paths must take into account the fact that some decision nodes may be part of the information state $I(j)$ of node $j \in D$, and, as such, will have their states observed (or fixed) in the set $S_{s_j|s_I(j)}$. Therefore, these decision nodes must be excluded from the product in the denominator in equation (14). Using (14), we can reformulate (13) into the strengthened form

$$\sum_{s \in S_{s_j|s_{I(j)}}} x(s) \leq \frac{|S_{s_j|s_I(j)}|}{\Pi_{k \in D \setminus (\{j\} \cup I(j))} |S_k|} z(s_j \mid s_{I(j)}), \quad \forall j \in D, s_j \in S_j, s_{I(j)} \in S_{I(j)}. \tag{15}$$

One last aspect that can be taken into account is that, depending on the problem structure, some sequence of states $s = (s_i)_{i=1,\ldots,n}$ forming a path may never be observed and can be preemptively filtered out from the set of paths S. This is the case, for example, in problems where earlier decisions or uncertain events dictate whether alternatives or uncertainties are observed. For instance, an initial decision regarding whether or not to build an industrial plant naturally restricts subsequent decisions regarding capacity expansion. Analogously, it may be that an uncertain production rate is only observed if one decides to build the production facility in the first place. To prevent the assembling of these unnecessary paths, we consider a set of *forbidden* paths, which, once removed, lead to a set $S^* \subseteq S$ of effective paths. Notice that these forbidden paths have probability zero by the structure of the problem, and therefore their removal does not affect the expected utility nor the constraints of the model. Furthermore, their removal allows for significant savings in terms of the scale of the model.

One issue emerges in settings where $S^* \subset S$ regarding the term (14). Notice that the bound is based on the premise that we can infer the total number of paths by considering the Cartesian product of the state sets S_j, $j \in N$. However, as forbidden paths are removed, some of the x-variables corresponding to paths $s \in S_{s_j|s_I(j)}$ might be removed,

making inequality (15) loose. A simple safeguard for this is to consider

$$\Gamma(s_j|s_{I(j)}) = \min\left\{|S^*_{s_j|s_{I(j)}}|, \frac{|S_{s_j|s_I(j)}|}{\Pi_{k\in D\setminus(\{j\}\cup I(j))}|S_k|}\right\} \tag{16}$$

and reformulate (15) as

$$\sum_{s\in S_{s_j|s_{I(j)}}} x(s) \leq \Gamma(s_j|s_{I(j)})z(s_j \mid s_{I(j)}), \quad \forall j \in D, s_j \in S_j, s_{I(j)} \in S_{I(j)}. \tag{17}$$

Combining the above, we can reformulate (5)–(10) as follows.

$$\underset{Z\in\mathbb{Z}}{\text{maximize}} \sum_{s\in S^*} U(s)p(s)x(s) \tag{18}$$

$$\text{subject to} \sum_{s_j\in S_j} z(s_j \mid s_{I(j)}) = 1, \qquad \forall j \in D, s_{I(j)} \in S_{I(j)} \tag{19}$$

$$\sum_{s\in S_{s_j|s_{I(j)}}} x(s) \leq \Gamma(s_j|s_{I(j)})z(s_j \mid s_{I(j)}), \quad \forall j \in D, s_j \in S_j, s_{I(j)} \in S_{I(j)} \tag{20}$$

$$\sum_{s\in S^*} p(s)x(s) = 1, \tag{21}$$

$$0 \leq x(s) \leq 1, \qquad \forall s \in S^* \tag{22}$$

$$z(s_j \mid s_{I(j)}) \in \{0, 1\}, \qquad \forall j \in D, s_j \in S_j, s_{I(j)} \in S_{I(j)}. \tag{23}$$

where $\Gamma(s_j|s_{I(j)})$ is defined as in (16). Note that this formulation preserves the (mixed-integer) linear nature of (5)–(10).

As discussed earlier, one of the main advantages of the decision programming formulation is the ability to incorporate objectives and constraints involving arbitrary utility functions and probability distributions within the model. Salo et al. [14] demonstrate this by proposing a model that considers conditional value-at-risk as one of the utility functions. The same can be achieved with our proposed formulation, by simply substituting $p(s)x(s)$ in place of variables $\pi(s)$.

The path-based structure of (5)–(10) makes formulating chance and budget constraints straightforward. For modeling chance constraints, we can define $\tilde{S}$ as the set of "undesirable" paths, the total probability of which must not exceed ρ. The corresponding chance constraint is then

$$\sum_{s\in\tilde{S}} x(s)p(s) \leq \rho. \tag{24}$$

Likewise, the set $\tilde{S}$ could further be defined as, e.g., the set of paths with a small utility $U(s) \leq u_{threshold}$. With $\rho = 0$, (24) can be seen as a budget constraint, stating

that for all compatible paths $s \in S(Z)$ with $p(s) > 0$, the utility $U(s)$ must be at least $u_{threshold}$.

4. Primal heuristic: single policy update (SPU)

While our approach of formulating influence diagrams into mixed-integer linear models does allow us to use powerful off-the-shelf solvers, it is still hindered by the well-known fact that solving such problems is NP-hard [15]. To make MIP solvers more efficient, *primal heuristics* are used to obtain and improve integer solutions. Obtaining good starting integer solutions can have a significant impact on the performance of branch-and-bound solvers, as it helps in pruning poor-quality solutions early.

Decision programming is based on limited-memory influence diagrams (LIMIDs) and solution approaches presented in previous literature can be used to obtain solutions to these problems. A notable contribution of Lauritzen and Nilsson [9] is the single policy update (SPU) heuristic for obtaining "locally optimal" strategies in the sense that the corresponding solutions cannot be improved by changing only one of the local strategies $Z_j(s_{I(j)})$.

Our proposed heuristic is loosely based on the ideas in Lauritzen and Nilsson [9], as described in Algorithm 1. The first step of the heuristic is to obtain a random strategy Z (note that this too is a heuristic, albeit a very simple one). Additionally, we initialize the $lastImprovement$ variable that will be used to stop the algorithm after finding a local optimum. The strategy Z is then iteratively improved by examining each information state $s_{I(j)} \in S_{I(j)}$ for each decision node $j \in D$ in order, choosing the local strategy $Z'_j(s_{I(j)})$ maximizing the expected utility. We obtain incrementally improving strategies by replacing the local strategy $Z_j(s_{I(j)})$ with $Z'_j(s_{I(j)})$ whenever the change results in an increase in expected utility. Finally, the pair $(j, s_{I(j)})$ is stored in the $lastImprovement$ variable if an improvement has been observed.

This process of locally improving the strategy is performed repeatedly for all pairs $(j, s_{I(j)})$ until no improvement is made during a whole iteration through the set of such pairs, that is, $(j, s_{I(j)}) = lastImprovement$. The number of possible strategies Z is finite, and the algorithm thus converges in a finite number of iterations. It is also easy to see that at termination, there is no possible local improvement and the strategy Z is thus, in that sense, locally optimal. Lauritzen and Nilsson [9] show that for *soluble* LIMIDs, this heuristic results in a globally optimal solution. However, influence diagrams are not

$Z \leftarrow randomstrategy()$;
$lastImprovement \leftarrow (undef, undef)$;
while *true* **do**
  for $j \in D$, $s_{I(j)} \in S_{I(j)}$ **do**
    if $(j, s_{I(j)}) = lastImprovement$ **then**
      return Z;
    else
      $Z'_j(s_{I(j)}) \leftarrow bestLocalStrategy(Z, j, s_{I(j)})$;
      $Z' \leftarrow modifyStrategy(Z, Z'_j(s_{I(j)}))$;
      if $EU(Z') > EU(Z)$ **then**
        $Z \leftarrow Z'$;
        $lastImprovement \leftarrow (j, s_{I(j)})$;
      end
    end
  end
end

Algorithm 1: The single policy update heuristic

generally soluble. The performance of the heuristic in DecisionProgramming.jl is explored in Section 6.

5. Julia interface

The ideas and formulations presented in this paper have been implemented as a Julia [3] package. The main purpose of the package is to provide an interface through which a user can build an influence diagram and automatically convert it to a MILP model that can be solved using an off-the-shelf solver. This brings us to the main reasons for choosing Julia as the language of the package. Thanks to JuMP [6] and MathOptInterface [10], implementing optimization problems in a general form is straightforward, and the models constructed in this way can be passed to many different solvers.

In order to create a user-friendly package that precludes a deeper understanding of MILP modeling by the user, we implemented a structure for influence diagrams and an interface for constructing them, demonstrated in Figure 3. In that, we illustrate the implementation of the pig farm problem, originally proposed in Lauritzen and Nilsson [9]. The influence diagram of the pig farm problem is presented in Figure 2.

The influence diagram structure consists of nodes and their state spaces and information sets, as well as the corresponding probabilities and utility values. The interface includes functions for adding these elements into the influence diagram and type structures that guide the user to include the required information. For example, all node

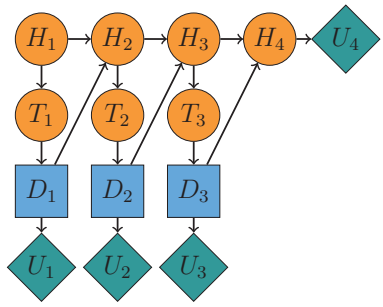


Figure 2: The influence diagram of the pig farm problem.

types require the user to define an information set for the node, even if it is empty (in the case of a root node). In turn, the function for adding nodes stops the user from accidentally including a node in its own information set, ensures that the names of all nodes are unique, and warns about redundant nodes.

```
using JuMP, Gurobi, DecisionProgramming

const N = 4

diagram = InfluenceDiagram()

# Syntax:
# ChanceNode(<name>, <information set>, <states>)
# DecisionNode(<name>, <information set>, <states>)
# ValueNode(<name>, <information set>)

add_node!(diagram, ChanceNode("H1", [], ["ill", "healthy"]))
for i in 1:N-1
    # Testing result
    add_node!(diagram, ChanceNode("T$i", ["H$i"], ["positive", "negative"]))
    # Decision to treat
    add_node!(diagram, DecisionNode("D$i", ["T$i"], ["treat", "pass"]))
    # Cost of treatment
    add_node!(diagram, ValueNode("C$i", ["D$i"]))
    # Health of next period
    add_node!(diagram, ChanceNode("H$(i+1)", ["H$(i)", "D$(i)"], ["ill", "healthy"]))
end
add_node!(diagram, ValueNode("MP", ["H$N"]))

generate_arcs!(diagram)
```

Figure 3: Code to generate the nodes in the pig farm problem [9]

Furthermore, specialised structures and functions (Figure 4) for defining and adding chance nodes' conditional probabilities and value nodes' utility values ensure that these matrices have correct dimensions, that the probabilities sum to one and that utility values are defined for all information states of value nodes. The interface also includes a function

for generating the arcs in the diagram and giving the nodes a topological order based on the information sets that the user has defined for them. This function also throws an error if the diagram has a directed cycle.

```
# Add probabilities for node H1
add_probabilities!(diagram, "H1", [0.1, 0.9])

# Declare probability matrix for health nodes H_2, ... H_N
X_H = ProbabilityMatrix(diagram, "H2")
X_H["healthy", "pass", :] = [0.2, 0.8]
X_H["healthy", "treat", :] = [0.1, 0.9]
X_H["ill", "pass", :] = [0.9, 0.1]
X_H["ill", "treat", :] = [0.5, 0.5]

# Declare probability matrix for test result nodes T_1...T_{N-1}
X_T = ProbabilityMatrix(diagram, "T1")
X_T["ill", "positive"] = 0.8
X_T["ill", "negative"] = 0.2
X_T["healthy", "negative"] = 0.9
X_T["healthy", "positive"] = 0.1

for i in 1:N-1
    add_probabilities!(diagram, "T$i", X_T)
    add_probabilities!(diagram, "H$(i+1)", X_H)
end

for i in 1:N-1
    add_utilities!(diagram, "C$i", [-100.0, 0.0])
end

add_utilities!(diagram, "MP", [300.0, 1000.0])

generate_diagram!(diagram)
```

Figure 4: Code to parametrize and generate an influence diagram

After the user has successfully generated an influence diagram, they can generate the model directly from the diagram structure, as shown in Figure 5. The model is generated using specialised functions for declaring decision variables and constraints which merely require the (empty) JuMP model and the influence diagram structure as parameters. These functions include optional keyword arguments which allow the user to define forbidden and fixed subpaths, probability cuts and probability scaling for better computational performance. A significant advantage of implementing the framework using JuMP is that more advanced users can easily extend these models by adding new variables or constraints as needed. All of the features presented in this paper are implemented in the framework. These include valid inequalities as lazy constraints, a conditional Value-at-Risk objective function and the single policy update heuristic.

Finally, the results of the optimization model can be extracted using a variety of functions. Figure 6 showcases arguably the most important of these, namely the functions

```
model = Model(Gurobi.Optimizer)
z = DecisionVariables(model, diagram)
x_s = PathCompatibilityVariables(model, diagram, z)
EV = expected_value(model, diagram, x_s)
@objective(model, Max, EV)
optimize!(model)
```

Figure 5: Code to solve the pig farm problem

for showing the optimal strategy. Other functions for analyzing results allow printing the distribution and statistics for the path utilities, and printing so-called state probabilities, that is, probabilities of given states occurring with the chosen strategy. For instance, in the pig farm problem, with the optimal strategy, the probability of the pig being healthy in stages 2, 3 and 4 is 73%, 70.5% and 69.5%, respectively.

```
Z = DecisionStrategy(z)
S_probabilities = StateProbabilities(diagram, Z)

print_decision_strategy(diagram, Z, S_probabilities)
# Output:
#
#  ┌─────────────────┬──────────────────┐
#  │ State(s) of T1  │ Decision in D1   │
#  ├─────────────────┼──────────────────┤
#  │ positive        │ pass             │
#  │ negative        │ pass             │
#  └─────────────────┴──────────────────┘
#
#  ┌─────────────────┬──────────────────┐
#  │ State(s) of T2  │ Decision in D2   │
#  ├─────────────────┼──────────────────┤
#  │ positive        │ treat            │
#  │ negative        │ pass             │
#  └─────────────────┴──────────────────┘
#
#  ┌─────────────────┬──────────────────┐
#  │ State(s) of T3  │ Decision in D3   │
#  ├─────────────────┼──────────────────┤
#  │ positive        │ treat            │
#  │ negative        │ pass             │
#  └─────────────────┴──────────────────┘
#
```

Figure 6: Code to obtain the optimal strategy for the pig farm problem

6. Computational experiments

We present a collection of computational experiments carried out to assess the performance of the proposed reformulation against the original formulation presented in Salo et al. [14]. Hereinafter, we use v1.2 to refer to our proposed model version as presented in (18)-(23) and v0.1 to refer to (5)-(10). These refer to the versions of `DecisionPrograming.jl` these formulations were available in.

In addition, we present computational results highlighting the performance of the proposed SPU heuristic. Those are tested considering (i) a modified version of the pig farm problem [9] in which we can artificially augment the number of time periods and generate input parameters randomly; (ii) the N-monitoring problem, as proposed in Salo et al. [14], in which we also can artificially augment the number of decision agents and randomly generate instances. All experiments are run using 16GB of memory and 8 threads on an Intel Xeon Gold 6248 CPU and the code can be found in https://github.com/gamma-opt/DecisionProgramming.jl.

6.1. Problem size

First, we compare the model sizes of the two formulations presented in Sections 2 and 3. In both formulations, the number of variables is the same. There are $\sum_{j \in D} |S_j||S_{I(j)}|$ z-variables and $|S|$ path variables, either π or x, depending on the formulation. As for the number of constraints, the formulation (5)-(10) has $\sum_{j \in D} |S_{I(j)}|$ constraints (6), $2|S|$ bounds for π-variables, $|D||S|$ constraints (8) and $|S|$ constraints (9). Arranging the terms, the total number of constraints becomes

$$(3 + |D|)|S| + \sum_{j \in D} |S_{I(j)}|. \tag{25}$$

The formulation (18)-(23) has $\sum_{j \in D} |S_{I(j)}|$ constraints (19), $\sum_{j \in D} |S_j||S_{I(j)}|$ constraints (20), one constraint (21) and $2|S|$ bounds for x-variables. Arranging the terms, the total number of constraints becomes

$$2|S| + \sum_{j \in D} (1 + |S_j|)|S_{I(j)}|. \tag{26}$$

We note that $|S| = \prod_{j \in C \cup D} |S_j|$ and that especially with a large number of nodes, the first term becomes impractically large in both (25) and (26). The increase in the number of path-related constraints is exponential, while the increase in the rest of the constraints is often linear, as shown in the following two example problems.

6.1.1. Pig farm

For the pig farm example presented in Lauritzen and Nilsson [9], we observe that the problem consists of $3n + 1$ decision and chance nodes, where n is the number of decision stages, and that $|S_j| = 2$ for all nodes $j \in C \cup D$. Note that this is slightly different to Lauritzen and Nilsson [9], where the length of the problem is tied to the number of health nodes. The length of a problem with n decision nodes would then be $n + 1$.

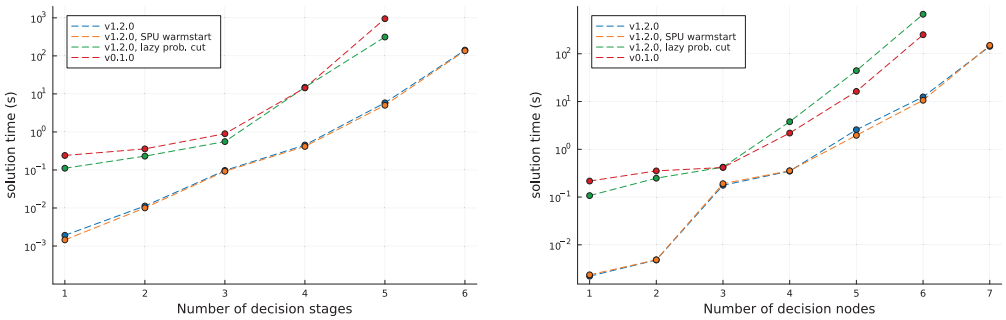
With these observations, $|S| = 2^{3n+1}$ and $|D| = n$. Thus, the number of constraints in Eq. (25) becomes $(3+n)2^{3n+1} + 2n$ and the corresponding number in Eq. (26) becomes $2^{3n+2} + 6n$.

6.1.2. N-monitoring

We can perform a similar analysis for the N-monitoring example presented in Salo et al. [14]. The problem consists of $2n+2$ decision and chance nodes, where n is the number of report-action pairs. As in the pig farm problem, $|S_j| = 2$ for all nodes $j \in C \cup D$.

With these observations, $|S| = 2^{2n+2}$ and $|D| = n$. Thus, the number of constraints in Eq. (25) becomes $(3+n)2^{2n+2} + 2n$ and the corresponding number in Eq. (26) becomes $2^{2n+3} + 6n$.

6.2. Solution times



(a) The pig farm problem

(b) The N-monitoring problem

Figure 7: The solution times of the two example problems with different number of decision nodes using different formulations. Notice the logarithmic y-axis.

Figure 7 shows the increase in average solution times over 50 instances as the number of decision stages increases in the two example problems. For the original formulation (5)-(10), Salo et al. [14] show that solution times are greatly improved by adding a *probability cut* $\sum_{s \in S} \pi(s) = 1$ as a lazy constraint to the model. A lazy constraint is a constraint that is added to the model formulation when it is deemed violated by an incumbent feasible solution found in the branch-and-cut tree search, instead of adding it from the beginning of the solution process. This approach is thus used in the computational experiments for the original formulation. For (18)-(23), a similar constraint is included in the formulation by default. However, we additionally analyze an instance of the the reformulated model (18)-(23), where constraint (21) is implemented as a lazy constraint.

For both problems, it seems that the rate of increase in the solution times quickly renders the original formulation (5)-(10) computationally intractable, as seen in Fig 7. This was also noted by Salo et al. [14] in their computational results. The solution times for the improved formulation (18)-(23) using locally compatible path sets seem to increase at a slower rate than for the original formulation. Finally, the lazy probability cut that was found to improve solution times in Salo et al. [14] is detrimental to computational performance in the new formulation (18)-(23).

	v0.1	v1.2
10th percentile	15.4	1.00
median	26.4	1.21
90th percentile	31.1	1.81
mean	25.0	1.34

Table 1: Statistics of the root relaxation quality relative to the optimal solution for 50 randomly generated pig farm problems with 5 decision stages. The solutions are scaled so that a value of 1 corresponds to the optimal solution.

In Table 1, we present statistics on the LP relaxation quality. As discussed before, the hypothesis is that the formulation (18)-(23) (implemented in v1.2 of the package) is considerably tighter than (5)-(10) (implemented in v0.1). The results from the pig farm problem strongly support this, as more than half of the LP relaxation solutions for the novel formulation are within 25% of the optimal solution, while the solutions using (5)-(10) are orders of magnitude further from the optimal solution.

Figure 8 shows the process of improving solutions in the single policy update (SPU) heuristic. For the 50 instances in this test set, the last solution is found within eight seconds, and the solution is the global optimum in all but one of the instances. Note that Lauritzen and Nilsson [9] showed that this version of the pig farm problem is not soluble, and thus the SPU heuristic is not guaranteed to find the optimal solution. We observe that while the single policy update heuristic is successful in finding good initial solutions quickly, the effect of providing the solver with these initial solutions is negligible (see Figure 7). Thus, our results suggest that improving the LP relaxation bound has a much greater impact on improving the solution time.

7. Case study: optimal preventive healthcare for CHD

One of the first frameworks for medical decision-making considering whether to treat, test or not treat was developed by Pauker and Kassirer [12]. This framework provides

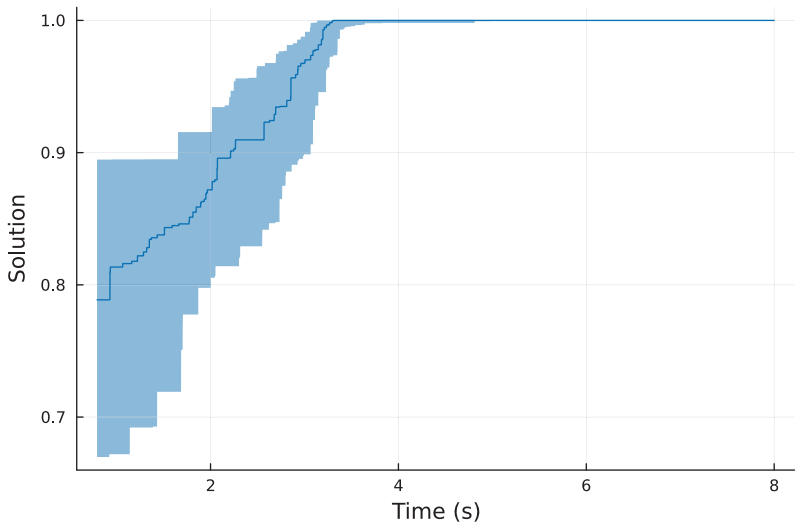


Figure 8: The median and first and third quartiles of solutions found by the SPU heuristic in 50 randomly generated pig farm problems with 6 decision stages. The solutions are scaled so that a value of 1 corresponds to the optimal solution.

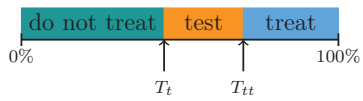


Figure 9: Testing (T_t) and test-treatment (T_{tt}) thresholds.

an analytical basis for optimal testing and treatment strategies. They developed two thresholds, referred to as "testing" and "test-treatment" thresholds. The thresholds are probability cut-offs and they divide subjects into three groups: if the risk of disease is below the "testing" threshold, treatment and testing should be withheld, if it is above the "test-treatment" threshold, treatment should be given and if the risk falls in between these thresholds then a diagnostic test should be performed and the treatment decision made based on its results. The thresholds are visualised in Figure 9.

In this case study we use decision programming to optimize the use of traditional and genetic testing to support the targeting of statin medication treatment for preventing coronary heart disease (CHD). This case study is replicated from Hynninen et al. [8], where the authors developed a testing and treatment strategy by optimizing net monetary benefit (NMB), a cost-benefit objective consisting of the health outcomes and testing costs within a 10-year time horizon.

The decision process stems from the patient's state of health, represented by a chance event H describing whether the patient will or will not have a CHD event in the following 10 years. The probability of a CHD event is assumed to be described by a prior risk estimate R_0 based on factors such as the age and sex of the patient. The likelihood of a correct prognosis can be improved by carrying out tests on traditional risk factors (TRS), genetic risk factors (GRS) or both. Based on their prognosis, a decision is made on whether a patient is subjected to preventive treatment with statin medication.

In Hynninen et al. [8], six predefined testing and treatment strategies were evaluated independently. In each of these strategies, the optimal allocation of tests and treatment according to risk estimates was obtained by solving the associated decision tree via dynamic programming. The six strategies considered in Hynninen et al. [8] were: (i) no tests and no treatment ('No treatment'); (ii) using prior risk to allocate treatment ('Treatment optimized'); (iii) performing TRS on optimized patient segment and allocating treatment based on updated risk estimates ('TRS optimized'); (iv) performing GRS on optimized patient segment and allocating treatment based on updated risk estimates ('GRS optimized'); (v) performing TRS on optimized patient segment and based on its results performing GRS optimally to allocate treatment ('TRS & GRS optimized'); (vi) performing GRS on optimized patient segment and based on its results performing TRS optimally to allocate treatment ('GRS & TRS optimized').

Essentially, this comprises determining optimal "testing" and "test-treatment" thresholds (cf. Figure 9) for TRS and GRS from the perspective of net monetary benefit (NMB) for each strategy (i-vi). Interestingly, the threshold values for GRS in Hynninen et al. [8] were different than the ones found in the study presented in Tikkanen et al. [16]. This is due to the different perspectives – pure patient welfare versus NMB – that the studies were conducted from. For example, the national health care guidelines for allocating treatment were not considered in the optimization in Hynninen et al. [8]. This showcases that the two thresholds described in Pauker and Kassirer [12] are not unique for a given disease and prognostic test because the perspective of the study affects the threshold values.

Analogously, our decision programming model determines an optimal decision strategy for allocating preventive care for CHD. The data and structure of the problem are the same as those utilised in Hynninen et al. [8]. However, due to the flexibility of decision

programming, the strategies (i-vi) do not need to be predefined. Instead, we can optimize the design of the strategy simultaneously with the threshold values, because all of these strategies are within the feasible solutions of the model.

The problem setting is such that the patient is assumed to have a prior risk estimate R_0. A risk estimate is a prediction of the patient's chance of having a CHD event in the next ten years. The risk estimates are grouped into risk levels, which range from 0% to 100% with a suitable discretization, e.g., $S_{R_0} = \{0\%, 1\%, ..., 99\%, 100\%\}$. We note that it might be beneficial to consider a less trivial discretization that is finer in the region where most of the probability mass is assumed to lie and coarser elsewhere. Nevertheless, we chose to proceed as such since it requires no information on the probability distributions. The first testing decision T_1 is made based on the prior risk estimate. This entails deciding whether to perform TRS or GRS or if no testing is needed. If a test is conducted, the risk estimate is updated (R_1) and based on the new information a second testing decision T_2 follows. It entails deciding whether further testing should be conducted or not. The second testing decision is constrained so that the same test which was conducted in the first stage cannot be repeated. If a second test is conducted, the risk estimate is updated again (R_2). The treatment decision T_D (dictating whether the patient receives preventive statin medicine or not) is made based on the resulting risk estimate of this testing process. Note that if no tests are conducted, the treatment decision is made based on the prior risk estimate. Figure 10 provides an influence diagram for the decision problem.

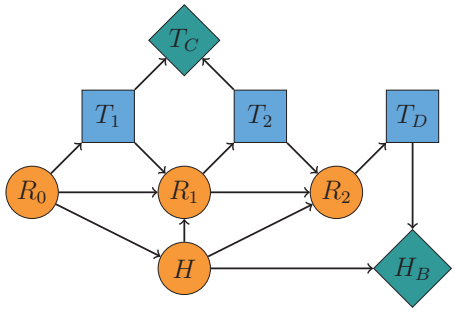


Figure 10: Influence diagram for the optimising the preventive care decision strategy for CHD.

Node H represents the uncertainty of whether the patient has a CHD event or remains healthy during the 10-year time frame. Node H has the prior risk level R_0 in its information set because a premise of the modeling proposed in Hynninen et al. [8] is that

the prior risk accurately describes the probability of having a CHD event, i.e.,

$$P(H = \text{CHD} \mid R_0 = \alpha) = \alpha.$$

On the other hand, nodes R_1 and R_2 represent the updated risk level after the first and second test decisions, respectively. If a test is conducted, the risk estimate is updated using the Bayes' rule

$$P(\text{CHD} \mid \text{test result}) = \frac{P(\text{test result} \mid \text{CHD}) \times P(\text{CHD})}{P(\text{test result})},$$

where the conditional probabilities $P(\text{test result} \mid \text{CHD})$ are from Abraham et al. [1] and the probability of having a CHD event, denoted by $P(\text{CHD})$, is the prior risk level R_0 or the updated risk level R_1, depending on whether it is the first or second test in question. The denominator $P(\text{test result})$ is calculated as a sum of the numerator and $P(\text{test result} \mid \text{no CHD}) \times P(\text{no CHD})$, where $P(\text{no CHD}) = 1 - P(\text{CHD})$. As the states of nodes R_i, $i \in \{0, 1, 2\}$, represent risk levels, the probability of a state in these nodes is the probability of the given test updating the risk estimate to that level from the previous estimate.

The first and second testing decisions are represented by T_1 and T_2, respectively. Since conducting the same test twice is forbidden, all paths where the same test is repeated in T_1 and T_2 are included in the set of forbidden paths (cf. Section 3). Furthermore, the forbidden paths include all paths where the first testing decision T_1 is to not perform testing but then the second testing decision T_2 is to perform a test. This is because the information yielded from performing only one test is not affected by whether the test is performed in the first or second stage of testing. Therefore, forbidding the paths where no test is performed in T_1 and a test is performed in T_2 reduces redundancy in the model without information loss. The final treatment decision is represented by node T_D, where the options are to provide or withhold treatment. The treatment decision is made based on the updated risk estimate represented by node R_2.

Since the first node in the influence diagram presented in Figure 10 is the chance node R_0, any decision strategy would be conditioned on its realisation. This leads to a natural separability of the problem, meaning that it can be solved for individual risk levels $0\%, 1\%, \ldots 100\%$. This has the benefit of allowing the calculations to be parallelised, at the expense of potentially causing inconsistencies related to e.g., multiple solutions in the MIP problem or rounding-induced errors.

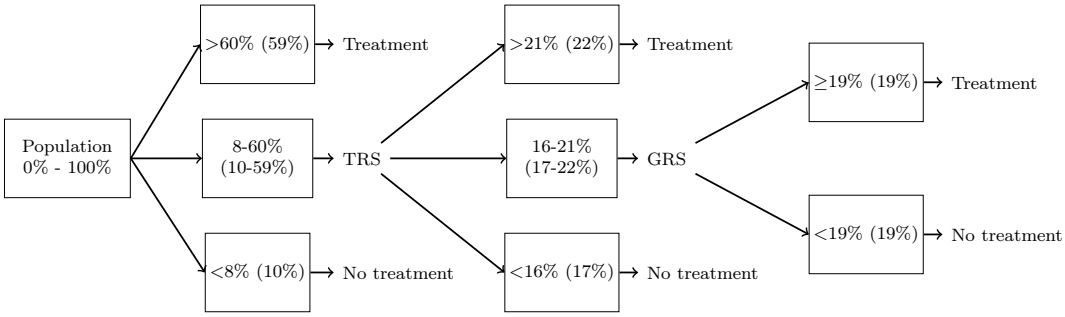


Figure 11: Optimal strategy obtained by our model (in parentheses, the original value from Hynninen et al. [8])

An interesting result is that the optimal strategy found by our model is the same strategy that was deemed the best among strategies (i-vi) in Hynninen et al. [8]. In a way, this provides optimality guarantees to their results, which, in principle, they could not have determined without exhaustively testing all possible (9) testing strategies. In addition, the optimal thresholds from our model correspond closely to those in Hynninen et al. [8]. Figure 11 illustrates the strategy obtained by our model, indicating also the thresholds found in Hynninen et al. [8] for comparison. We are confident that the small differences in the threshold values are simply artefacts related to the way the discretization (i.e., rounding) was performed.

8. Conclusions

In this paper, we expand on the ideas originally proposed in Salo et al. [14] providing multiple methodological enhancements. These enhancements include a novel and more efficient formulation, valid bounds to tighten relaxations, and a heuristic which can be used to find feasible solutions and, consequently, to warm start the MIP solver. We also introduce `DecisionProgramming.jl`, a Julia package that allows representing decision problems as MIP models.

Furthermore, we conduct a novel case study based on the study originally proposed by Hynninen et al. [8]. Our objective is to demonstrate that our package can be used in settings which would normally require resorting to more ad-hoc computational tools and that it lends itself to be a general and accessible tool for practitioners. We believe that this intuitive and accessible interface, using `JuMP.jl`, a state-of-the-art mathemat-

ical programming modeling package, will allow for a much wider range of practitioners and researchers to have access to mathematical optimization-based tools for supporting decision-making. Furthermore, this will create novel inroads for the use of mathematical optimization in the area of decision analysis at large, potentially unveiling new and promising directions for future developments.

In terms of alternative further developments, we see several directions that deserve further investigation. First, decision programming as a modeling framework is still in its infancy, and, consequently, many obstacles are still to be overcome for its widespread adoption. One of these obstacles is computational requirements. Decision programming models grow large as the number of nodes and/or states increase, and thus it would greatly benefit from alternative ideas that can tackle such large-scale problems. These can be, for example, related to alternative formulations that convert the influence diagram into an intermediate structure and employ ideas from Bayesian inference to yield a more compact MIP model (see Parmentier et al. [11]). Another direction worth exploring is the employment of decomposition methods, in particular, those which allow for a delayed generation of structural elements of the model, in our case the paths $s \in S$ (see Section 2). Another interesting avenue would be to pursue methods that can reap benefits from employing parallelization, given the increasing availability of high-performance computing clusters.

Acknowledgements

We are enormously grateful for the input from Juho Andelmin, whose initial implementations led to the development of `DecisionProgramming.jl`. We are also grateful for the contributions of a number of graduate and undergraduate students to the development of the package, as well as to the welcoming and supportive JuMP community. We also gratefully acknowledge the financial support from the Research Council of Finland (decision number 332180). Finally, the computational experiments were performed using computer resources within the Aalto University School of Science "Science-IT" project.

References

[1] Gad Abraham, Aki S Havulinna, Oneil G Bhalala, Sean G Byars, Alysha M De Livera, Laxman Yetukuri, Emmi Tikkanen, Markus Perola, Heribert Schunkert, Eric J

Sijbrands, et al. Genomic prediction of coronary heart disease. *European Heart Journal*, 37(43):3267–3278, 2016.

[2] Dimitris Bertsimas and Jack Dunn. *Machine learning under a modern optimization lens*. Dynamic Ideas LLC Charlestown, MA, 2019.

[3] Jeff Bezanson, Alan Edelman, Stefan Karpinski, and Viral B Shah. Julia: A fresh approach to numerical computing. *SIAM Review*, 59(1):65–98, 2017.

[4] Concha Bielza, Manuel Gómez, and Prakash P Shenoy. A review of representation issues and modeling challenges with influence diagrams. *Omega*, 39(3):227–241, 2011.

[5] John R Birge and Francois Louveaux. *Introduction to stochastic programming*. Springer Science & Business Media, 2011.

[6] Iain Dunning, Joey Huchette, and Miles Lubin. JuMP: A Modeling Language for Mathematical Optimization. *SIAM Review*, 59(2):295–320, 2017.

[7] Ronald A Howard and James E Matheson. Influence diagrams. *Decision Analysis*, 2(3):127–143, 2005.

[8] Yrjänä Hynninen, Miika Linna, and Eeva Vilkkumaa. Value of genetic testing in the prevention of coronary heart disease events. *PloS one*, 14(1):e0210010, 2019.

[9] Steffen L Lauritzen and Dennis Nilsson. Representing and solving decision problems with limited information. *Management Science*, 47(9):1235–1251, 2001.

[10] Benoît Legat, Oscar Dowson, Joaquim Dias Garcia, and Miles Lubin. Mathoptinterface: a data structure for mathematical optimization problems. *INFORMS Journal on Computing*, 34(2):672–689, 2022.

[11] Axel Parmentier, Victor Cohen, Vincent Leclère, Guillaume Obozinski, and Joseph Salmon. Integer programming on the junction tree polytope for influence diagrams. *INFORMS Journal on Optimization*, 2(3):209–228, 2020.

[12] Stephen G Pauker and Jerome P Kassirer. The threshold approach to clinical decision making. *New England Journal of Medicine*, 302(20):1109–1117, 1980.

[13] Martin L Puterman. Markov decision processes. *Handbooks in operations research and management science*, 2:331–434, 1990.

[14] Ahti Salo, Juho Andelmin, and Fabricio Oliveira. Decision programming for mixed-integer multi-stage optimization under uncertainty. *European Journal of Operational Research*, 299(2):550–565, 2022.

[15] Alexander Schrijver et al. *Combinatorial optimization: polyhedra and efficiency*, volume A. Springer, 2003.

[16] Emmi Tikkanen, Aki S Havulinna, Aarno Palotie, Veikko Salomaa, and Samuli Ripatti. Genetic risk prediction and a 2-stage risk screening strategy for coronary heart disease. *Arteriosclerosis, thrombosis, and vascular biology*, 33(9):2261–2266, 2013.

Publication II

Olli Herrala, Topias Terho, Fabricio Oliveira. Risk-averse decision strategies for influence diagrams using rooted junction trees. Submitted to *Operations Research Letters*, December 2023.

Risk-averse decision strategies for influence diagrams using rooted junction trees

Olli Herrala[a], Topias Terho[a], Fabricio Oliveira[a,*]

[a]*Department of Mathematics and Systems Analysis, Aalto University, School of Science, FI-00076 Aalto, Finland*

Abstract

This paper focuses on a mixed-integer programming formulation for influence diagrams, based on a gradual rooted junction tree representation of the diagram. We show that different risk considerations, including chance constraints and conditional value-at-risk, can be incorporated into the formulation with targeted, appropriate modifications to the diagram structure. The computational performance of the formulation is assessed on two example problems and is found to be highly dependent on the structure of the junction tree.

Keywords: influence diagram, mixed-integer programming, risk-aversion

1. Introduction

Influence diagrams (ID) [11] are an intuitive structural representation of decision problems with uncertainties and interdependencies between random events, decisions and consequences. Traditional solution methods for influence diagrams [25] often require strong assumptions such as the no-forgetting assumption. Lauritzen and Nilsson [16] present the notion of limited memory influence diagrams (LIMID) that, albeit more general in terms of representation capabilities, do not satisfy the no-forgetting assumption and, therefore, are not amenable to these traditional methods.

The algorithms presented in the literature for solving decision problems represented as IDs are mostly suited only to problems where an expected utility function is maximized and no additional constraints are considered. Thus, often risk considerations are encoded in the utility function itself, by making it concave using, e.g., utility extraction techniques [4, 20, 8].

Very often, utility functions represent monetary values, such as costs or revenues. In that case, maximizing expected utility assumes a risk-neutral stance from the decision-maker. However, decision-makers may still have different risk tolerance profiles, which must be represented in the decision process.

There are numerous ways to incorporate risk aversion into decision models without requiring utility extraction techniques. A typical method is to minimize a risk measure instead of expected utility [18]. A commonly used measure is the Conditional Value-at-Risk (CVaR), which measures the expected loss value in the α-tail, α being a confidence level parameter [22]. Another typical way of incorporating risk aversion is to use constraints such as those related to chance events or budget violations [2]. Both mentioned methods have been used widely in various applications (See, e.g., [6, 27, 13]). The main challenge with all of these is that they connect different decisions so that methods based on local computations (e.g., decision trees) cannot be straightforwardly employed.

Recently, two different mixed-integer programming (MIP) reformulations for influence diagrams have emerged, likely stemming from the considerable computational improvements in MIP solution methods. The reformulation considered in this paper is originally presented in Parmentier et al. [21], where the authors first show how to convert a LIMID representing an expected utility maximization problem into a gradual rooted junction tree. This junction tree consists of clusters of nodes from the LIMID and is reformulated as a MIP problem using marginal probability distributions of nodes within each cluster.

In contrast, Salo et al. [23] present decision programming, which directly reformulates a LIMID as a mixed-integer linear programming (MILP) formulation without the intermediate clustering step of forming a junction tree. The main advantage of decision programming is that its formulation can be adapted to minimize (conditional) value-at-risk, and its path-based MILP formulation makes it easy to consider different constraints, as discussed in Hankimaa et al. [10].

Comparing the two approaches, the clustering step employed in Parmentier et al. [21] generally results in considerably improved computational performance compared to decision programming. Against this backdrop, this paper presents an approach to incorporate the risk measures and constraints from Salo et al. [23] and Hankimaa et al. [10] in the rooted junction tree reformulation proposed by Parmentier et al. [21]. This allows us to enjoy the modelling flexibility of Decision Programming while reaping the computational benefits of the junction tree reformulation.

In Section 2, we present background on (limited memory) influence diagrams and the MIP reformulations of such diagrams. Section 3 continues with extending the rooted junction tree-based reformulation to consider different risk measures and constraints, demonstrated in two example problems in Section 4. Finally, Section 5 concludes the paper with ideas on future research directions and the potential of reformulating influence

*Corresponding author: fabricio.oliveira@aalto.fi

diagrams as MIP problems.

2. Background

2.1. Pig farm problem

The pig farm problem is an example of a partially observable Markov decision process (POMDP) and is used throughout this paper as the running example to illustrate the proposed developments. Cohen and Parmentier [3] further discuss the modelling of POMDPs using the methodology from Parmentier et al. [21], but we keep our focus on the more general formulations presented in the latter.

In the pig farm problem [16], a farmer is raising pigs for a period of four months after which the pigs will be sold. During the breeding period, a pig may develop a disease, which negatively affects the retail price of the pig at the time they are sold. In the original formulation, a healthy pig commands a price of 1000 DKK and an ill pig commands a price of 300 DKK. During the first three months, a veterinarian visits the farm and tests the pigs for the disease. The specificity (or true negative rate) of the test is 80%, whereas the sensitivity (true positive rate) is 90%. Based on the test results, the farmer may decide to inject a medicine, which costs 100 DKK. The medicine cures an ill pig with a probability of 0.5, whereas an ill pig that is not treated is spontaneously cured with a probability of 0.1. If the medicine is given to a healthy pig, the probability of developing the disease in the subsequent month is 0.1, whereas the probability without the injection is 0.2. In the first month, a pig has the disease with a probability of 0.1.

2.2. Influence Diagrams

An influence diagram is a directed acyclic graph $G = (N, A)$, where N is the set of nodes and A is the set of arcs. Let $N = N^C \cup N^D \cup N^V$ be the set of chance nodes N^C, value nodes N^V and decision nodes N^D in the influence diagram. Let $I(j)$, $j \in N$, denote the information set (also often called parents) of j, i.e., nodes from which there is an arc to j. The influence diagram of the pig farm problem is presented in Figure 1.

Each node $j \in N$ has a discrete and finite state space S_j representing possible outcomes. The outcome (i.e., state) s_j of a stochastic node $m \in N^C \cup N^V$ is a random variable with a probability distribution $\mathbb{P}(X_m = s_m \mid X_{I(m)} = s_{I(m)})$, where the notation $X_j = s_j$ means that the node(s) j attain the state(s) s_j. The states of a value node $v \in N^V$ represent different outcomes that have a utility value $u(s_v)$ associated with them. The outcome of a decision node $d \in N^D$ is determined by a *decision strategy* $\delta(s_d \mid s_{I(d)}) : S_{I(d)} \cup S_d \to \{0, 1\}$.

The solution of an influence diagram is a decision strategy that optimizes the desired metric, typically expected utility, at value nodes. A common additional assumption is perfect recall, meaning that previous decisions can be recalled in later stages. Under this assumption, the optimal decision strategy may be obtained by arc reversals and node removals [24] or dynamic programming [26], for example.

Perfect recall is a rather strict assumption and in many applications, it does not hold. This challenge is circumvented with limited memory influence diagrams [16]. Many algorithms for solving the decision strategy that maximizes the expected utility have been developed, such as the single policy update [16], multiple policy updating [17], branch and bound search [12] and the aforementioned methods converting the influence diagram to a MI(L)P [21, 23].

2.3. Rooted Junction Trees

An influence diagram $G = (N, A)$ can be represented as a directed rooted tree $\mathscr{G} = (\mathscr{V}, \mathscr{A})$ composed of *clusters* $C \in \mathscr{V}$, which are subsets of the nodes of the ID, that is, $C \subset N$. Both G and $\mathscr{G}$ are directed acyclic graphs whose vertices are connected with directed arcs in A and $\mathscr{A}$, respectively. The main difference between these diagrams lies in the nature of the vertices. In an influence diagram, the set of nodes N consists of individual chance events, decisions and consequences, while the clusters in $\mathscr{V}$ comprise multiple nodes, hence the notational distinction between N and $\mathscr{V}$.

In order to reformulate this tree into a MIP model, we impose additional conditions, making $\mathscr{G}$ a *gradual rooted junction tree*. Definition 2.1 states the necessary properties of a gradual rooted junction tree.

Definition 2.1. *A directed rooted tree $\mathscr{G} = (\mathscr{V}, \mathscr{A})$ consisting of clusters $C \in \mathscr{V}$ of nodes $j \in N$ is a gradual rooted junction tree corresponding to the influence diagram G if*

(a) given two clusters C_1 and C_2 in the junction tree, any cluster C on the unique undirected path between C_1 and C_2 satisfies $C_1 \cap C_2 \subset C$;

(b) each cluster $C \in \mathscr{V}$ is the root cluster of exactly one node $j \in N$ (that is, the root of the subgraph induced by the clusters with node j) and all nodes $j \in N$ appear in at least one of the clusters;

(c) and, for each cluster, $I(j) \in C_j$, where C_j is the root cluster *of $j \in N$.*

A rooted tree satisfying part (a) in Definition 2.1 is said to satisfy the *running intersection property*. This condition is sufficient for making $\mathscr{G}$ a rooted junction tree (RJT). In addition, as a consequence of part (b), we see that a gradual RJT has as many clusters as the original influence diagram has nodes, and each node $j \in N$ can be thought as corresponding to one of the clusters $C \in \mathscr{V}$. Because of this, we refer to clusters using the corresponding nodes $j \in N$ in the influence diagram as the *root cluster* of node $j \in N$, which is denoted as $C_j \in \mathscr{V}$.

Formulating an optimization model based on the gradual RJT representation starts by introducing a vector of moments μ_{C_j} for each cluster C_j, $j \in N$. Parmentier et al. [21] show that for RJTs, we can impose constraints so that these become moments of a distribution μ_N that factorizes according to $G(N, A)$. The joint distribution $\mathbb{P}$ is said to factorize [14] according to G if $\mathbb{P}$ can be expressed as

$$\mathbb{P}(X_N = s_N) = \prod_{j \in N} \mathbb{P}(X_j = s_j \mid X_{I(j)} = s_{I(j)}). \tag{1}$$

In the formulation, $\mu_{C_j}(s_{C_j})$ represents the probability of the nodes within the cluster C_j being in states s_{C_j} and part (c) of Definition 2.1 ensures that $\mathbb{P}(X_j = s_j \mid X_{I(j)} = s_{I(j)})$ can thus be obtained from $\mu_{C_j}(s_{C_j})$ for each $j \in N$. The resulting MIP model is

$$\max \sum_{j \in N^V} \sum_{s_{C_j} \in S_{C_j}} \mu_{C_j}(s_{C_j}) u_{C_j}(s_{C_j}) \tag{2}$$

$$\text{s.t.} \sum_{s_{C_j} \in S_{C_j}} \mu_{C_j}(s_{C_j}) = 1, \ \forall j \in N \tag{3}$$

$$\sum_{\substack{s_{C_i} \in S_{C_i}, \\ s_{C_i \cap C_j} = s^*_{C_i \cap C_j}}} \mu_{C_i}(s_{C_i}) = \sum_{\substack{s_{C_j} \in S_{C_j}, \\ s_{C_i \cap C_j} = s^*_{C_i \cap C_j}}} \mu_{C_j}(s_{C_j}),$$

$$\forall (C_i, C_j) \in \mathscr{A}, s^*_{C_i \cap C_j} \in S_{C_i \cap C_j} \tag{4}$$

$$\mu_{C_j}(s_{C_j}) = \mu_{\overline{C}_j}(s_{\overline{C}_j}) p(s_j \mid s_{I(j)}), \ \forall j \in N^C \cup N^V, s_{C_j} \in S_{C_j} \tag{5}$$

$$\mu_{C_j}(s_{C_j}) = \mu_{\overline{C}_j}(s_{\overline{C}_j}) \delta(s_j \mid s_{I(j)}), \ \forall j \in N^D, s_{C_j} \in S_{C_j} \tag{6}$$

$$\mu_{C_j}(s_{C_j}) \geq 0, \ \forall j \in N, s_{C_j} \in S_{C_j} \tag{7}$$

$$\delta(s_j \mid s_{I(j)}) \in \{0, 1\}, \ \forall j \in N^D, s_j \in S_j, s_{I(j)} \in S_{I(j)}. \tag{8}$$

The formulation (2)-(8) is an expected utility maximization problem where u_{C_j} in the objective function (2) represents the utility values associated with different realizations of the nodes within the cluster C_j, and $\overline{C}_j = C_j \setminus j$ is used in constraints (5) and (6) for notational brevity. Constraints (3) and (7) state that the variables μ_{C_j} must represent valid probability distributions, with nonnegative probabilities summing to one.

Constraint (4) enforces local consistency between adjacent clusters, meaning that for a pair C_i, C_j of adjacent clusters, the marginal distribution for the nodes in both C_i and C_j (that is, $C_i \cap C_j$) must be the same when obtained from either C_i or C_j.

To ease the notation, moments $\mu_{\overline{C}_j}(s_{\overline{C}_j}) = \sum_{s_j \in S_j} \mu_{C_j}(s_{C_j})$ are used in constraints (5) and (6). The expression $\mu_{\overline{C}_j}$ represents the marginal distribution for cluster C_j with the node j marginalized out. The value $p(s_j \mid s_{I(j)})$ is the conditional probability of a state s_j given the information state $s_{I(j)}$ and $\delta(s_j \mid s_{I(j)})$ the decision strategy in node $j \in N^D$. It should be noted that constraint (6) involves a product of two variables, and is thus not linear. Since we are limiting ourselves to settings with deterministic strategies (i.e., $\delta(s_d \mid s_{I(d)}) : S_{I(d)} \cup S_d \to \{0, 1\}$), these constraints become indicator constraints and can be efficiently handled by solvers such as Gurobi [9]. We remark that this would not be the case for more general strategies of the form $\delta(s_d \mid s_{I(d)}) : S_{I(d)} \cup S_d \to [0, 1]$.

Any rooted tree satisfying the properties in Definition 2.1 is a gradual RJT and can therefore be used to obtain a valid version of model (2)-(8). For instance, a junction tree where each cluster C_j, $j \in N$, contains the nodes $i \in N$ such that $i \leq j$ would satisfy the definition. However, this would result in large clusters, and consequently, a large number of constraints (4)-(6). It is thus important to find a gradual RJT representation where the clusters are as small as possible. Parmentier et al. [21] present two algorithms for creating a gradual RJT from an ID. The first algorithm uses a given topological order of the nodes and builds the RJT starting from the last cluster and proceeding in the reverse direction of this topological order. This algorithm returns a gradual RJT with minimal clusters given the ordering of nodes. The second algorithm has an additional step of finding a "good" topological order that would lead to smaller clusters. The contribution of this paper is focused on modifying the underlying influence diagram to which both algorithms can be applied. For simplicity, we chose to use the algorithm requiring a topological order in the examples of this paper. Using $H_1, T_1, D_1, V_1, H_2, ..., H_4, V_4$ as a topological order, the pig farm influence diagram in Figure 1 is transformed to the gradual RJT in Figure 2.

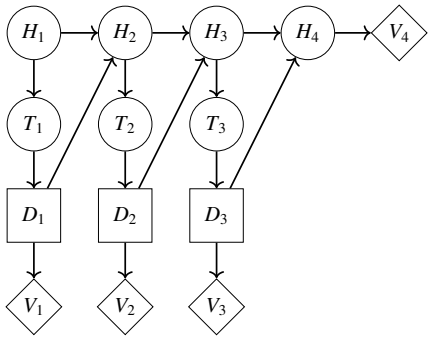


Figure 1: The pig farm problem [16].

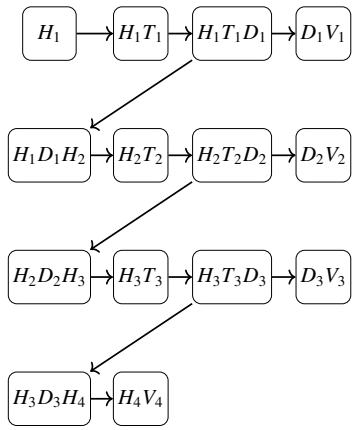


Figure 2: Gradual RJT of the pig farm problem.

3. Our contributions

3.1. Extracting the utility distribution

For problems with multiple value nodes, e.g., multi-stage decision problems, the expected utility has the convenient property that the total expected utility is the sum of expected utilities in each value node. This property can be exploited in the solution process, and for this reason, many solution methods for influence diagrams, including the RJT approach in Parmentier et al. [21], only tackle maximum expected utility (MEU) problems.

In contrast, risk measures (such as CVaR) require that the full probability distribution of the consequences is explicitly represented in the model. However, such representations are lost when the value nodes are placed in separate clusters, as in Figure 2, since the probability distributions are only defined for each cluster separately. For example, in the pig farm problem described in Section 2.1, the joint distribution of V_1 and V_2 cannot be inferred from the probability distributions of clusters C_{V_1} and C_{V_2}, as we cannot assume the probabilities of consequences in V_1 and V_2 to be independent.

We note that after solving the model (2)-(8), any distribution can be obtained for the MEU solution. As stated in Definition 2.1, part (c), the rooted junction trees in this paper have $I(j) \subseteq C_j$ by construction, and we can thus use the optimal values $\bar{\mu}_j$ to obtain $\mathbb{P}(X_j = s_j \mid X_{I(j)} = s_{I(j)})$ for all nodes $j \in N$, and consequently use Eq. 1 to obtain $\bar{\mu}_N$. Then, we can obtain the marginal distribution for nodes $N_1 \subset N$ by marginalizing out $N \setminus N_1$. More formally,

$$\mathbb{P}(s^*_{N_1}) = \sum_{s_N \in \{S_N \mid s_{N_1} = s^*_{N_1}\}} \bar{\mu}_N.$$

Using this approach for incorporating constraints and objectives involving the utility distribution of multiple value nodes in (2)-(8) would require obtaining the distribution μ_N within the model. This, however, would require products of arbitrarily many continuous variables μ_{C_j} within the model, resulting in nonlinearity and the associated computational challenges.

The issue can be circumvented by modifying the influence diagram such that the consequences of the problem are represented by a single value node. Generally, multiple value nodes represent components of a separable utility function such that $U(s) = \sum_{v \in N^V} U_v(s_{I(v)})$ [26] and, being such, the value nodes can be combined under a single value node $\bar{v}$, in which the consequences can simply be evaluated with $U(s_{I(\bar{v})})$.

This transformation requires that arcs $(p, \bar{v}), \forall v \in N^V, p \in I(v)$, are added to A. Then, according to Definition 2.1, part (c), we have that $\cup_{v \in V} I(v) \subset C_{\bar{v}}$. Consequently, the marginal probability distribution $\mu_{C_{\bar{v}}}$ contains information on the joint probability $\mathbb{P}(\cap_{v \in N^V} v)$ and this can be exposed to produce a probability distribution for utility values. Following this approach, the modified influence diagram of the modified pig farm problem is presented in Figure 3 and the corresponding gradual RJT in Figure 4.

This however, incurs in computationally more demanding versions of model (2)-(8). In the modified pig farm problem, all nodes D_k are in the information set of $\bar{V}$, and it follows from the running intersection property that D_k must be contained in every cluster that is in the undirected path between C_{D_k} and $C_{\bar{v}}$. Therefore, the clusters become larger as the parents of value nodes are "carried over", instead of evaluating separable components of the utility function at different value nodes. As will be discussed in Section 3.4, this transformation comes with a cost on the computational efficiency.

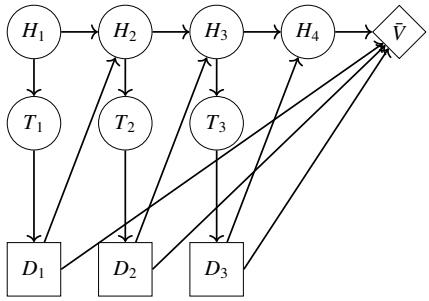


Figure 3: The pig farm problem reformulated [16].

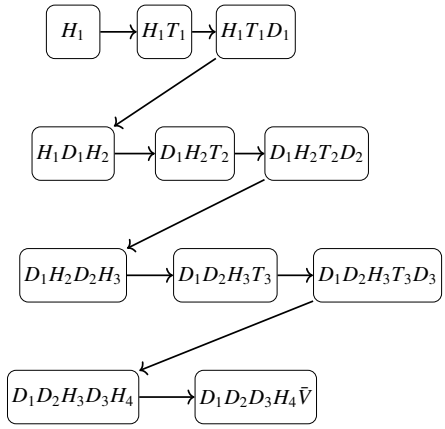


Figure 4: Gradual RJT of the reformulated pig farm problem.

3.2. Imposing chance, logical, and budget constraints

The proposed reconfiguration of the influence diagram allows us to expose the vector of moments of the value node $\bar{v}$, which in turn, enables the formulation of a broad range of risk-aversion-related constraints.

For example, a chance constraint can be constructed based on the utility distribution from the vector of moments of the value node $\bar{v}$ as:

$$\sum_{s_{C_{\bar{v}}} \in \{S_{C_{\bar{v}}} \mid s_{\bar{v}} \in S^o_{\bar{v}}\}} \mu(s_{C_{\bar{v}}}) \leq p, \tag{9}$$

where $S^o_{\bar{v}}$ is the set of outcomes that the decision maker wishes to constrain and $p \in [0, 1]$. For instance, assume that a decision

maker wishes to add chance constraints enforcing that the probability of the payout of the process being less than some fixed limit l is at most p. Then $S^o_{\bar{v}}$ would contain all states $s_{\bar{v}}$ such that $U(s_{\bar{v}}) < l$.

Chance constraints on the probability distribution of a single node can be straightforwardly added to any cluster containing the node. For instance, chance constraints enforcing that a node n must be in state s^*_n with a probability greater or equal than p can be formulated as

$$\sum_{s_{C_n} \in \{S_{C_n} | s_n = s^*_n\}} \mu_C(s_{C_n}) \geq p.$$

Note that this formulation can be enforced for any clusters C_k such that $n \in C_k$. Then, to keep the number of variables in the constraint to a minimum, one can choose the smallest of such clusters, i.e., choose cluster $k' = \arg\min_{k \in N}\{|C_k|,\ n \in C_k\}$.

Logical constraints can be seen as a special case of chance constraints. For example, in the pig farm problem (in Section 2), the farmer may wish to attain an optimal decision strategy while ensuring that the number of injections is at most two per pig due to a limited availability of injections. Then, $S^o_{\bar{v}}$ would contain all realizations of the nodes in $C_{\bar{v}}$ that would lead to a violation of the constraint, i.e., the state combinations in which three injections would be given to a pig. Then, constraint (10) that makes these scenarios impossible could be imposed, i.e.,

$$\sum_{s_{C_{\bar{v}}} \in \{S_{C_{\bar{v}}} | s_{\bar{v}} \in S^o_{\bar{v}}\}} \mu(s_{C_{\bar{v}}}) \leq 0. \tag{10}$$

Budget constraints are analogous to logical constraints, as the farmer could instead have an injection budget, say 200 DKK per pig. Then, $S^o_{\bar{v}}$ should contain all states $s_{\bar{v}}$, where more than 200 DKK is used for treating a pig, with the constraint enforced similarly as in (10).

3.3. Conditional Value-at-Risk

In addition to a number of utility distribution-related constraints, a single value node also enables the consideration of alternative risk measures. Next, we focus our presentation on how to maximize conditional value-at-risk. However, we highlight that other risk metrics such as absolute or lower semi-absolute deviation [23] can, in principle, be used. The entropic risk measure [7] can also be used as a constraint. However, incorporating it in the objective function is likely to introduce nonlinearity in the model due to the logarithmic nature of the measure.

The proposed formulation for conditional value-at-risk maximization is analogous to the method developed for decision programming in [23]. Denote the possible utility values with $u \in U$ and suppose we can define the probability $p(u)$ of attaining a given utility value. In the presence of a single value node, we would define $p(u) = \sum_{s_{C_v} \in \{S_{C_v} | U(s_{C_v}) = u\}} \mu(s_{C_v})$. We can then pose the constraints

$$\eta - u \leq M\lambda(u), \qquad \forall u \in U \tag{11}$$
$$\eta - u \geq (M + \epsilon)\lambda(u) - M, \qquad \forall u \in U \tag{12}$$
$$\eta - u \leq (M + \epsilon)\overline{\lambda}(u) - \epsilon, \qquad \forall u \in U \tag{13}$$
$$\eta - u \geq M(\overline{\lambda}(u) - 1), \qquad \forall u \in U \tag{14}$$
$$\overline{\rho}(u) \leq \overline{\lambda}(u), \qquad \forall u \in U \tag{15}$$
$$p(u) - (1 - \lambda(u)) \leq \rho(u) \leq \lambda(u), \qquad \forall u \in U \tag{16}$$
$$\rho(u) \leq \overline{\rho}(u) \leq p(u), \qquad \forall u \in U \tag{17}$$
$$\sum_{u \in U} \overline{\rho}(u) = \alpha \tag{18}$$
$$\lambda(u), \overline{\lambda}(u) \in \{0, 1\}, \qquad \forall u \in U \tag{19}$$
$$\rho(u), \overline{\rho}(u) \in [0, 1], \qquad \forall u \in U, \tag{20}$$
$$\eta \in \mathbb{R} \tag{21}$$

where α is the probability level in VaR_α. The constraints force the values of the decision variables to the values in Table 1.

variable	value
η	VaR_α
$\lambda(u)$	1 if $u < \eta$
$\overline{\lambda}(u)$	0 if $u > \eta$
$\rho(u)$	0 if $\lambda(u) = 0$, $p(u)$ otherwise
$\overline{\rho}(u)$	$\begin{cases} p(u) & \text{if } u < \eta, \\ \alpha - \sum_{u \in U} p(u) & \text{if } u = \eta, \\ 0 & \text{if } u > \eta\ (\overline{\lambda}(u) = 0) \end{cases}$

Table 1: Variables and the corresponding values that satisfy (11)-(20)

In constraints (11)-(20), M is a large positive number and ϵ is a small positive number. The parameter ϵ is used to model strict inequalities, which generally cannot be directly used in mathematical optimization solvers. For example, $x \geq \epsilon$ is assumed to be equivalent to $x > 0$. When $\lambda(u) = 0$, constraints (11) and (12) become $-M \leq \eta - u \leq 0$, or $\eta \leq u$. When $\lambda(u) = 0$, they instead become $\epsilon \leq \eta - u \leq M$, or $\eta > u$. Constraints (13) and (14) can be examined similarly to obtain the results in Table 1.

The correct behavior of variables $\rho(u)$ is enforced by (16) and (17). If $\lambda(u) = 0$, constraint (16) forces $\rho(u)$ to zero. If $\lambda(u) = 1$, $\rho(u) = p(u)$. Finally, assuming η is equal to VaR_α, now that we have $\lambda(u) = 1 \longrightarrow \rho(u) = p(u)$ for all $u < \eta$, the value of $\overline{\rho}(u)$ has to be $\alpha - \sum_{u \in U} \rho(u)$ for $u = \eta$. It is easy to see that η must be equal to VaR_α for there to be a feasible solution for the other variables. For a more rigorous proof, see Salo et al. [23, Appendix A].

By introducing the constraints above to the optimization model, $CVaR_\alpha$ can then be obtained as $\frac{1}{\alpha} \sum_{u \in U} \overline{\rho}(u)u$. This can be either used as in the objective function or as a part of the constraints of the problem. We also note that the described approach is very versatile in that u can be selected to be, e.g., a stage-specific utility function, thus allowing us to limit risk in specific stages of a multi-stage problem. Krokhmal et al. [15] discusses the implications of stage-wise CVaR-constraints in detail.

3.4. Problem size

From Definition 2.1, we can derive a relationship between the width of the tree and the size of the corresponding model. By definition, a tree with a width k has a maximum cluster C_n containing $k+1$ nodes. In a gradual RJT, the cluster C_n includes exactly one node $n \in N$ not contained in its parent cluster C_i. Using the running intersection property, the k other nodes in C_n must also be in C_i. If we make the very light assumption that all nodes $l \in C_i \cup C_n$ have at least two states s_l, this implies that there are $\prod_{l \in C_i \cup C_n} |S_n| \geq 2^k$ local consistency constraints (4) for the pair $(C_i, C_n) \in \mathscr{A}$ and the number of constraints in the model (2)-(8) is thus at least $O(c^k)$, where k is the width of the gradual RJT. This is in line with Parmentier et al. [21] pointing out that the RJT-based approach is only suited for problems with moderate rooted treewidth.

The width of the tree in Figure 4 is $N + 1$, where N is the number of treatment periods in the pig farm problem ($N = 3$ in the example), while the width of the original pig farm RJT in Figure 2 is only 2. Furthermore, we note that the rooted treewidth of a problem is defined as the size of the largest cluster minus one. In an RJT, we have $(I(n) \cup n) \subset C_n$ for all $n \in N$ and the treewidth is thus at least $max(|I(n)|,\ n \in N)$. For the single value node pig farm problem, $|I(\bar{V})| = N + 1$ and for the original pig farm problem, $|I(H_2)| = 2$. Therefore, we conclude that there are no RJT representations for these problems with a smaller width than the ones presented in Figures 2 and 4.

These results imply that the optimization model for the original pig farm problem grows linearly with the number of stages, but both the single value node pig farm grows exponentially with the number of decisions, suggesting possible computational challenges in larger problems.

4. Computational experiments

To assess the computational performance of the model (2)-(8), we use the pig farm problem described earlier and the N-monitoring problem from Salo et al. [23]. Both problems are solved with varying numbers of decision nodes, providing insights into the growth in solution times with increasing problem sizes. The models were implemented using Julia v1.7.3 [1] and JuMP v1.5.0 [5] and solved with the Gurobi solver v10.0.0 [9].

4.1. Pig farm problem

A six-month pig farm problem (five treatment periods and a final selling period) is used to highlight the use of the developed formulations. The constraints presented in Section 3.3 are added to the optimization model so that the problem can be optimized taking into account the CVaR of the chosen solution.

This enables determining nondominated strategies based on CVaR and expected utility values using, for example, the epsilon constraint method [19]. The example in Figure 5 shows the nondominated strategies based on expected utility and CVaR with probability level $\alpha = 0.05$ (orange points) and a sample of the dominated strategies (blue points).

The nondominated strategies from highest expected utility to lowest are:

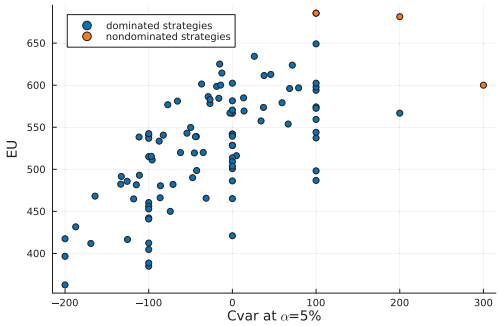


Figure 5: Nondominated strategies

- Treat the pig in the 4th and 5th period if the test result is positive
- Treat the pig in the 5th period regardless of the test result
- Never treat the pig

Using the formulations presented in Section 3.2 the six-month pig farm problem can also be solved with a variety of chance constraints. For instance, we might be interested in a decision strategy that maximizes the expected utility while ensuring the following:

- A pig is healthy in the last period with a probability of 80% or higher;
- The payout is at least 800 DKK with a probability greater or equal to 50%.

The decision strategy is then to treat the pig in the 3rd period if the test result is positive and in the 4th and 5th periods no matter the test result. This way the expected utility is 627 DKK.

4.2. N-monitoring

The N-monitoring problem [23] represents a problem of distributed decision-making where N decisions are made in parallel with no communication between the decision-makers. The node L in Figure 6 represents a load on a structure, nodes R_i are reports of the load, based on which the corresponding fortification decisions A_i are made. The probability of failure in node F depends on the load and the fortification decisions, and the utility in T comprises fortification costs and a reward if the structure does not fail.

With topological order $L, R_1, A_1, ..., R_N, A_N, F, T$, the rooted junction tree corresponding to the diagram in Figure 6 is presented in Figure 7. The structure of parallel observations and decisions in the N-monitoring problem is very different compared to the partially observed Markov decision process (POMDP) structure of the pig farm problem. From Figure 6, we can see that $|I(F)| = N + 1$, and consequently, there are no RJT representations for the N-monitoring problem with a width less than $N + 1$. In contrast, as discussed in Section 3.4, the pig

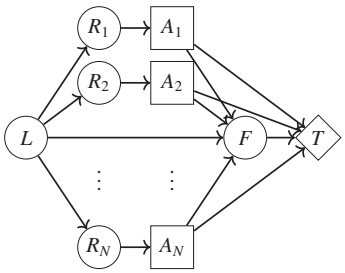


Figure 6: An influence diagram representing the N-monitoring problem.

farm RJT has a constant width of 2, independent of the number of decision stages. We note that the width of the RJT in Figure 7 is the same as of the single value node pig farm RJT (Figure 4). However, in the N-monitoring problem, this is a consequence of the inherent structure of the problem, instead of the influence diagram manipulation described in Section 3.1. That is, for the pig farm problem, there is a small width RJT representation for MEU problems, but such representations are impossible for the N-monitoring problem due to its structure.

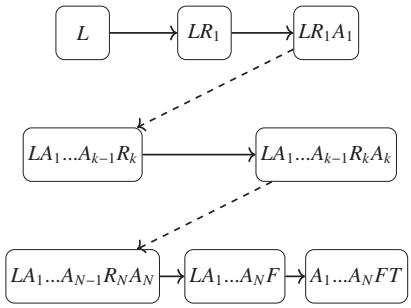


Figure 7: A rooted junction tree representing the N-monitoring problem.

4.3. Computational results

Figure 8 shows the increase in solution times as the problem size increases. First, we see that the solution time for the N-monitoring problem increases the fastest. While the pig farm problem with a single value node is faster than the N-monitoring problem, the solution time seems to also increase exponentially. Finally, the solution time for the pig farm problem as presented in Section 2.1 does not change significantly. These results are in line with the analysis of model sizes in Section 3.4 and highlight the importance of keeping the cluster sizes small in junction trees. Both versions of the pig farm problem compared in Figure 8 solve exactly the same problem, but the structure of the diagram in the single value node problem increases the width of the tree, and subsequently, the model size and solution time.

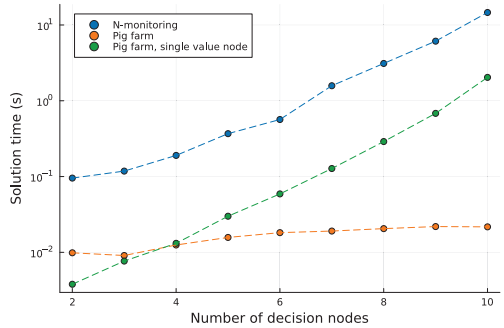


Figure 8: Mean solution times for 50 random instances in the pig farm and N-monitoring problems with 2-10 decision nodes.

Finally, we compare these results to the corresponding results using decision programming, presented in Hankimaa et al. [10]. In the N-monitoring problem and the single value node pig farm problem, we see similar exponential growth for both models, with the solution times in this paper being 2-3 orders of magnitude smaller. For the original pig farm problem, the decision programming formulation remains the same, as the model size is determined by the chance and decision nodes in the diagram. However, for this version of the problem, the solution times for (2)-(8) hardly even change with the model sizes tested in this section, illustrating the superior computational efficiency of the model with small treewidths.

5. Conclusions

In this paper, we have described a MIP reformulation of decision problems presented as (limited memory) influence diagrams, originally proposed in Parmentier et al. [21]. Our main contribution is to extend the modelling framework from Parmentier et al. [21] to embed it with more general modelling capabilities. We illustrate how, e.g., chance constraints and conditional value-at-risk can be incorporated into the formulation.

We also present some results on the relationship between the rooted treewidth of the RJT representation and the size of the corresponding MIP model, along with solution times from two different decision problems. The pig farm problem is a partially observed Markov decision process (POMDP) and very large instances can be solved to optimality within seconds. The N-monitoring problem, on the other hand, is an example of distributed decision-making, where N decision-makers must make decisions in parallel with no communication between them. We found that, for such problems, the size of the formulation (2)-(8) grows exponentially with N, resulting in the solution times becoming considerably large for N as small as 10.

We find that the model presented in Parmentier et al. [21] can be extended beyond pure expected utility maximization problems to incorporate most of the constraints and objective func-

tions present in decision programming, the alternative MIP reformulation based on LIMIDs described in Salo et al. [23] and Hankimaa et al. [10]. The advantage of using the models described in this paper is that in terms of model size, decision programming models grow exponentially with respect to the number of nodes, which seems to be only the worst-case behaviour with rooted junction trees. Inspecting the formulation (2)-(8), we notice that the number of constraints is mainly affected by the local consistency constraints (4), as the number of all other constraints is linear in the number of nodes. The number of constraints for the pig farm RJT in Figure 2 is $O(N)$, where N is the number of decision stages. On the other hand, the same formulation for the N-monitoring RJT in Figure 7 has $O(2^N)$ constraints, exponential in the number of parallel decisions. For a worst-case example, in a diagram with arcs (i, j) for each $i, j \in N, i < j$ (semicomplete digraph), the number of constraints would be exponential in the number of nodes.

In this paper, the extraction of relevant probability/utility distributions is made possible by modifying the underlying influence diagram. For future research, it might be beneficial to note that any gradual RJT (Definition 2.1) can be used to formulate the MIP model (2)-(8). Notably, it should be possible to modify the RJT so that relevant nodes are "carried over" to, e.g., the last cluster, giving us access to the joint probability distributions required for the models described in this paper.

Acknowledgements

Funding: This work was supported by the Academy of Finland [*Decision Programming: A Stochastic Optimization Framework for Multi-Stage Decision Problems*, grant number 332180]. We are also thankful for the contributions from Prof. Ahti Salo.

References

[1] Bezanson, J., Edelman, A., Karpinski, S., Shah, V.B., 2017. Julia: A fresh approach to numerical computing. SIAM Review 59, 65–98.

[2] Charnes, A., Cooper, W., 1959. Chance constrained programming. Management Science 6, 73–79.

[3] Cohen, V., Parmentier, A., 2023. Future memories are not needed for large classes of POMDPs. Operations Research Letters 51, 270–277.

[4] Davidson, D., Suppes, P., Siegel, S., 1957. Decision making; an experimental approach. Decision making; an experimental approach., Stanford University Press. Pages: 121.

[5] Dunning, I., Huchette, J., Lubin, M., 2017. JuMP: A Modeling Language for Mathematical Optimization. SIAM Review 59, 295–320.

[6] Filippi, C., Guastaroba, G., Speranza, M., 2020. Conditional value-at-risk beyond finance: a survey. International Transactions in Operational Research 27, 1277–1319.

[7] Föllmer, H., Knispel, T., 2011. Entropic risk measures: Coherence vs. convexity, model ambiguity and robust large deviations. Stochastics and Dynamics 11, 333–351.

[8] Geissel, S., Sass, J., Seifried, F.T., 2018. Optimal expected utility risk measures. Statistics & Risk Modeling 35, 73–87.

[9] Gurobi Optimization, LLC, 2022. Gurobi Optimizer Reference Manual. URL: `https://www.gurobi.com`.

[10] Hankimaa, H., Herrala, O., Oliveira, F., Tollander de Balsch, J., 2023. Decisionprogramming.jl – a framework for modelling decision problems using mathematical programming. `arXiv:2307.13299`.

[11] Howard, R.A., Matheson, J.E., 2005. Influence diagrams. Decision Analysis 2, 127–143.

[12] Khaled, A., Hansen, E.A., Yuan, C., 2013. Solving limited-memory influence diagrams using branch-and-bound search, in: Uncertainty in Artificial Intelligence (UAI-13), AUAI Press, Arlington, Virginia, USA. pp. 331–341.

[13] Khassiba, A., Bastin, F., Cafieri, S., Gendron, B., Mongeaua, M., 2020. Two-stage stochastic mixed-integer programming with chance constraints for extended aircraft arrival management. Trasnportation Science 54, 897–919.

[14] Koller, D., Friedman, N., 2009. Probabilistic graphical models: principles and techniques. MIT press.

[15] Krokhmal, P., Palmquist, J., Uryasev, S., 2002. Portfolio optimization with conditional value-at-risk objective and constraints. Journal of risk 4, 43–68.

[16] Lauritzen, S.L., Nilsson, D., 2001. Representing and solving decision problems with limited information. Management Science 47, 1235–1251.

[17] Mauà, D.D., de Campos, C.P., Zaffalon, M., 2012. Solving limited memory influence diagrams. Journal of Artificial Intelligence Research 44, 97–140.

[18] Homem-de Mello, T., Pagnoncelli, B.K., 2016. Risk aversion in multistage stochastic programming: A modeling and algorithmic perspective. European Journal of Operational Research 249, 188–199.

[19] Miettinen, K.M., 1999. Nonlinear multiobjective optimization. Kluwer Academic Publishers.

[20] Nielsen, T.D., Jensen, F.V., 2004. Learning a decision maker's utility function from (possibly) inconsistent behavior. Artificial Intelligence 160, 53–78.

[21] Parmentier, A., Cohen, V., Leclère, V., Obozinski, G., Salmon, J., 2020. Integer programming on the junction tree polytope for influence diagrams. INFORMS Journal on Optimization 2, 209–228.

[22] Rockafellar, R., Uryasev, S., 2000. Optimization of conditional value-at-risk. Journal of Risk 2, 21–42.

[23] Salo, A., Andelmin, J., Oliveira, F., 2022. Decision programming for mixed-integer multi-stage optimization under uncertainty. European Journal of Operational Research 299, 550–565.

[24] Shachter, R.D., 1986. Evaluating influence diagrams. Operations Research 34, 871–882.

[25] Shachter, R.D., Bhattacharjya, D., 2010. Solving influence diagrams: Exact algorithms. Wiley encyclopedia of operations research and management science .

[26] Tatman, J.A., Shachter, R.D., 1990. Dynamic programming and influence diagrams. IEEE Transactions on Systems, Man, and Cybernetics 30, 365–379.

[27] Xu, B., Boyce, S.E., Zhang, Y., Liu, Q., Guo, L., Zhong, P.A., 2017. Stochastic programming with a joint chance constraint model for reservoir refill operation considering flood risk. Journal of Water Resources Planning and Management 143.

Publication III

Olli Herrala, Tommi Ekholm, Fabricio Oliveira. Solving decision problems with endogenous uncertainty and conditional information revelation using influence diagrams. Submitted to *Omega*, February 2024.

Solving decision problems with endogenous uncertainty and conditional information revelation using influence diagrams

Olli Herrala[a], Tommi Ekholm[b], Fabricio Oliveira[a,*]

[a]*Department of Mathematics and Systems Analysis, Aalto University, School of Science, FI-00076 Aalto, Finland*
[b]*Finnish Meteorological Institute, Helsinki, Finland*

Abstract

Despite methodological advances for modeling decision problems under uncertainty, representing endogenous uncertainty still proves challenging both in terms of modeling capabilities and computational requirements. A novel reformulation based on rooted junction trees (RJTs) provides an approach for solving such decision problems using off-the-shelf mathematical optimization solvers. This is made possible by using influence diagrams to represent a given decision problem and reformulating them as RJTs, which are then represented as a mixed-integer linear programming problem.

In this paper, we focus on the type of endogenous uncertainty that received less attention in the rooted junction tree approach: conditionally observed information. Multi-stage stochastic programming models use conditional non-anticipativity constraints to represent such uncertainties, and we show how such constraints can be incorporated into RJT models. This allows us to consider the two main types of endogenous uncertainty simultaneously, namely decision-dependent information structure and decision-dependent probability distribution. Finally, the extended framework is illustrated with a large-scale cost-benefit problem regarding climate change mitigation.

Keywords: endogenous uncertainty, stochastic programming, junction trees, climate change mitigation

1. Introduction

Stochastic programming (SP) is one of the most widespread mathematical programming-based frameworks for decision-making under uncertainty. In general, SP casts decision problems subject to parametric uncertainty as deterministic equivalents in the form of large-scale linear or mixed-integer programming (LP/MIP) problems that can be solved with standard optimization techniques. A common assumption in SP models is that the stochastic processes, particularly the state probabilities and/or observed values, are not influenced by the previously made decisions. The uncertainty is thus *exogenous*. This is methodologically convenient, for the deterministic equivalent model has the same nature as its stochastic counterpart, retaining important characteristics such as linearity, or more generally, convexity.

In this paper, we focus on a much less explored class of stochastic problems presenting *endogenous* uncertainty. In this more general setting, the decisions made at previous stages can affect the uncertainty faced in later stages. It is common to classify SP problems according to the nature of the endogenous structure arising in the decision problem. Hellemo et al. (2018) propose a taxonomy of such problems, classifying the endogenous uncertainties into two distinct types. In Type 1 problems, earlier decisions influence the later events' probability distribution (i.e.,

*Corresponding author: fabricio.oliveira@aalto.fi

realizations and/or the probabilities associated with each realization). For example, deciding to perform maintenance on a car engine influences the probability of it breaking in the future. In Type 2 problems, the *information structure* is influenced by the decision-making. Continuing with the car example, deciding to inspect the engine does not affect the probability of it breaking, but provides information that enables a better-informed maintenance decision. It is noteworthy that Type 2 is more common in the literature on decision-making under endogenous uncertainty. Hellemo et al. (2018) also introduce Type 3, which combines the Type 1 and Type 2 endogenous uncertainties.

Our main contribution is to provide a framework general enough to address both types simultaneously, resulting in Type 3 endogenously uncertain SP (T3ESP) problems, while still retaining prior computationally favorable properties of the mathematical model such as linearity or convexity. Our proposed framework builds upon the work by Parmentier et al. (2020) and Herrala et al. (2023) and expands the models to consider Type 2 endogenous uncertainty. A major advantage of this approach is the ability to incorporate Type 1 endogenous uncertainties in the decision process in an intuitive way by using an influence diagram representation of the problem. This representation is then reformulated into a rooted junction tree and further converted to a mixed-integer linear programming (MILP) problem, for which powerful off-the-shelf solvers exist. The intermediate step is not strictly necessary, as Salo et al. (2022) present an alternative formulation directly converting the influence diagram into a MILP. However, as discussed in Herrala et al. (2023), the RJT representation results in better computational performance. Salo et al. (2022) present a simple problem with Type 2 endogenous uncertainty using their Decision Programming framework but do not discuss this class of endogenous uncertainty in detail. This paper explores Type 2 endogenous uncertainty further, showing explicitly how the RJT reformulation can be enhanced to become a suitable framework for T3ESP problems comprising both Type 1 and 2 endogenous uncertainties.

The formulation presented in Parmentier et al. (2020) accommodates only discrete decisions, and Salo et al. (2022) acknowledge the limitations of using influence diagram-based formulations for problems involving continuous decisions, discussed in more detail in Bielza et al. (2011). In addition to discussing T3ESP problems, we show that if the problem has a separable structure, it is possible to incorporate continuous decisions that do not affect the endogenous probabilities in the model. This is demonstrated in a climate change mitigation case study, but we note that similar structures are likely to arise in other problems as well. Indeed, Lee et al. (2021) show how such structures can be used in a submodel-tree decomposition for influence diagrams. In the case study that originally motivated our developments, we consider uncertain climate parameters and technological progress and the problem of determining the optimal strategy for climate and technology research, as well as the optimal emission levels for 2030-2070.

This paper is structured as follows. In Section 2, we present an overview of multi-stage stochastic programming. In Section 3, our methodological contributions are described in detail, starting from the formulation in Parmentier et al. (2020) and continuing with conditionally observed information. In Section 4 we illustrate the use of the framework by considering a larger-scale problem of climate change cost-benefit analysis. Section 5 concludes and provides directions for further development.

2. Modeling problems with endogenous and exogenous uncertainties

Solution approaches for MSSP are often based on formulating the deterministic equivalent problem using a scenario tree, as described in, e.g., Ruszczyński (1997). A scenario tree represents the structure of the uncertain decision process, and non-anticipativity constraints (NACs) (Rockafellar and Wets, 1991) are employed to enforce the information structure in the formulation. NACs state that a decision must be the same for two scenarios if those scenarios are indistinguishable when making the decision.

In endogenously uncertain problems, the decisions can affect the timing, event probabilities, or outcomes of uncertain events further in the process. As previously discussed, endogenous uncertainty is often divided into *decision-dependent probabilities* (Type 1) and *decision-dependent information structure* (Type 2) (Hellemo et al., 2018). In this context, information structure often refers to when the realization of each uncertain event is observed, if ever. In contrast, exogenously uncertain problems have a fixed information structure with the timing of observations known a priori.

Type 2 uncertainty has been more widely addressed in the SP literature, perhaps due to one of its subclasses having a strong connection to exogenously uncertain problems. In the taxonomy presented in Hellemo et al. (2018), this specific type of endogenous uncertainty is called *conditional information revelation*. In this subclass, the decisions only affect the time at which the (exogenous) uncertainty is revealed to the decision maker. One of the earliest publications on such uncertainty is Jonsbråten et al. (1998), where the authors describe a branching algorithm for solving a subcontracting problem. Goel and Grossmann (2006) consider a process network problem where the yield of a new process is uncertain prior to installation. Other applications include open pit mining (Boland et al., 2008), clinical trial planning for drug development (Colvin and Maravelias, 2010) and technology project portfolio management (Solak et al., 2008).

Similar solution methods are employed in problems with exogenous uncertainty and conditionally revealed information. The main difference is that conditional information revelation requires the use of *conditional non-anticipativity constraints* (C-NACs), as the distinguishability is dependent on earlier decisions. This conditional dependency results in disjunctive constraints that require specific reformulation techniques (Apap and Grossmann, 2017). The main challenge arising from this approach is that the number of constraints rapidly increases with problem size, resulting in computational intractability for large problems. Apap and Grossmann (2017) also propose omitting redundant constraints, in an attempt to mitigate the tractability issues. In their example production planning problem, this results in roughly a 99% decrease in the problem size. Despite these substantial improvements, the reduced model is still very large and cannot be solved to the optimum within a reasonable computation time under their experimental setting, illustrating how challenging such problems are.

In addition to conditional information revelation, Type 2 uncertainty also encompasses problems where the information structure is altered by modifying the support of decision variables and changing the objective or constraint coefficients. Hellemo et al. (2018) report a 2SSP problem where the recourse costs depend on first stage decisions (Ntaimo et al., 2012). Gustafsson and Salo (2005) present a model for contingent portfolio programming, a project scheduling problem where projects can be expanded or terminated before they are finished. The decisions thus affect the decision spaces of future decision variables, resulting in additional consistency constraints, e.g., that a project can be continued in period t only if it was ongoing in period $t-1$.

Type 1 endogenous uncertainty is more challenging from a mathematical modeling standpoint because the uncertain events depend on earlier decisions, which, in turn, precludes a scenario tree-based representation as the scenario probabilities in a tree cannot depend on decisions. Therefore, the well-established solution techniques for MSSP cannot be directly applied to these problems. Despite these challenges, some discussion on Type 1 endogenous uncertainty is found in the literature. Peeta et al. (2010) discuss the fortification of a structure in a network, where the probability of failure depends on the fortification decision. Lauritzen and Nilsson (2001) present the "pig farm problem", where the health of a pig depends on the treatment decisions, introducing the concept of limited memory influence diagrams (LIMID), thus relaxing the no-forgetting assumption in influence diagrams. The no-forgetting assumption states that when making a decision, all prior decisions and outcomes of uncertain events are known. This assumption results in significant limitations for distributed decision-making and

leads to computational challenges in multi-period problems where the decisions towards the end are conditional on the full history of the problem. Dupačová (2006) presents a summary of problems with Type 1 uncertainties and Escudero et al. (2020) present solution approaches to multi-stage problems where the first-stage decisions influence the scenario probabilities in later stages. Examples of such problems can be found in Zhou et al. (2022) and Li and Liu (2023). Finally, reformulations and custom algorithms for various Type 1 problems are further summarized in Hellemo et al. (2018). However, these approaches assume specific relationships between decisions and probabilities and result in non-convex nonlinear formulations. Consequently, these approaches are not easily generalizable to different problems.

In this paper, we present a general solution framework for problems with both types of endogenous uncertainty discussed here. Modeling Type 1 endogenous uncertainty has previously relied on specific problem structures and reformulations, and the ability to model these uncertainties in a general setting makes the proposed framework versatile compared to these earlier methods. Additionally, we present two alternative approaches for incorporating Type 2 endogenous uncertainty in the decision models formulated with our framework. The ease of considering any combination of the two types of endogenous uncertainty also makes the framework more generally applicable to problems with endogenous uncertainty than the approaches discussed earlier.

3. Rooted junction trees with conditionally observed information

3.1. Influence diagrams

An *influence diagram* $G_{ID} = (N, A_{ID})$ is an acyclic graph formed by nodes $j \in N = N^C \cup N^D \cup N^V$ and arcs $a \in A_{ID} = \{(i,j) \mid i,j \in N\}$[1]. Nodes N^C and N^D are the sets of chance and decision nodes, respectively, and N^V is a collection of value nodes representing the consequences incurred from the decisions made at nodes N^D and the chance events realized at nodes N^C. In Fig. 1, the decision nodes are represented by squares, the chance nodes by circles, and the value nodes by diamonds.

Each decision and chance node $j \in N^C \cup N^D$ can assume a state s_j from a discrete and finite set of states S_j. For a decision node $j \in N^D$, S_j represents the available choices; for a chance node $j \in N^C$, S_j is the set of possible realizations. The arcs (i,j) in $A_{ID} = \{(i,j) \mid i,j \in N\}$ represent influence between nodes. In Fig. 1, the arcs are represented by arrows between the nodes. Before defining this notion of influence further, let us first define a few necessary concepts.

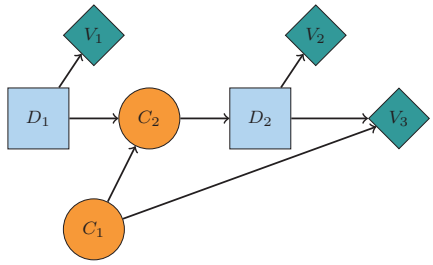


Figure 1: An influence diagram representation of a decision problem

The *information set* comprises the immediate predecessors of a given node $j \in N$ and is defined as $I(j) = \{i \in N \mid (i,j) \in A_{ID}\}$. In the graphical representation, this corresponds to

[1]We use the subscript ID for G_{ID} and A_{ID} to distinguish between influence diagrams and rooted junction trees. Rooted junction trees are described in Section 3.2.

the set of nodes that have an arrow pointing directly to node j. For example, in Fig. 1, the information set of C_2 consists of D_1 and C_1. The decision $s_j \in S_j$ made in each decision node $j \in N^D$ and the conditional probabilities of the states $s_j \in S_j$ in each chance node $j \in N^C$ depend on their *information state* $s_{I(j)} \in S_{I(j)}$, where $S_{I(j)} = \prod_{i \in I(j)} S_i$. Referring back to our example, the probabilities of different outcomes in C_2 are conditional on the decision in D_1 and the random outcome in C_1. Let us define $X_j \in S_j$ as the realized state at a chance node $j \in N^C$. Using the notion of information states, the conditional probability of observing a given state s_j for $j \in N^C$ is $\mathbb{P}(X_j = s_j \mid X_{I(j)} = s_{I(j)})$.

For a decision node $j \in N^D$, let $Z_j : S_{I(j)} \to S_j$ be a mapping between each information state $s_{I(j)} \in S_{I(j)}$ and decision $s_j \in S_j$. That is, $Z_j(s_{I(j)})$ defines a *local decision strategy*, which represents the choice of some $s_j \in S_j$ in $j \in N^D$, given the information $s_{I(j)}$. Note that we do not consider mixed strategies, where each information state would be mapped to an arbitrary probability distribution over S_j. Instead, we only consider deterministic strategies that can be represented by an indicator function $\mathbb{I} : S_{I(j)} \times S_j \to \{0, 1\}$ defined so that

$$\mathbb{I}(s_{I(j)}, s_j) = \begin{cases} 1, & \text{if } Z_j \text{ maps } s_{I(j)} \text{ to } s_j, \text{ i.e., } Z_j(s_{I(j)}) = s_j; \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

A (global) *decision strategy* is the collection of local decision strategies in all decision nodes: $Z = (Z_j)_{j \in N^D}$, selected from the set of all possible strategies $\mathbb{Z}$.

At the value node $v \in N^V$, a real-valued utility function $U_v : S_{I(v)} \to \mathbb{R}$ maps the information state $s_{I(v)}$ of v to a utility value U_v. The default objective is to maximize the expected utility of a strategy, but other objectives such as conditional Value-at-Risk can also be used (Salo et al., 2022; Herrala et al., 2023).

3.2. Rooted junction trees

As shown in Salo et al. (2022), it is possible to obtain a mixed-integer linear programming (MILP) model directly from the influence diagram representation of the problem. However, the authors observe that the model size increases exponentially with the number of nodes, resulting in computational challenges with relatively small problems. To mitigate this exponential growth, Parmentier et al. (2020) proposes first reformulating the influence diagram into a *rooted junction tree* (RJT) $G_{RJT} = (V, A_{RJT})$, a directed graph consisting of clusters $C \in V$ of nodes $j \in N$, and arcs between these clusters, with the underlying undirected graph (obtained by replacing the directed edges A_{RJT} with undirected edges) being a tree. The first important property of an RJT is the *running intersection property*, i.e., if a node $j \in N$ is in two clusters of the tree, it is also in all clusters on the (undirected) path between these clusters. For example, in Fig. 2, the node D_1 appears in the four leftmost clusters. From this property, it follows that the subgraph of G_{RJT} induced by a node j (formed by clusters $C \in V$ for which $j \in C$, and the arcs connecting such clusters) is a rooted tree.

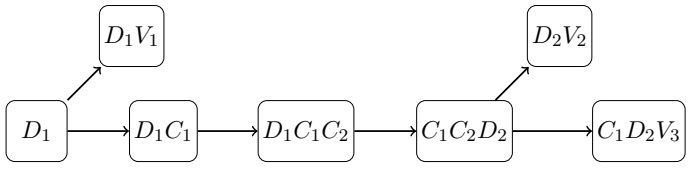


Figure 2: A gradual RJT representation of the ID in Fig. 1

More specifically, Parmentier et al. (2020) consider gradual RJTs, where each cluster is the *root cluster* of exactly one node $j \in N$, and the root cluster of node j is denoted with C_j. A root cluster is defined as the root of the subgraph of the RJT induced by a node j. For example, the

subgraph induced by D_1 consists of the four leftmost clusters in Fig. 2 and the arcs connecting them. The root of this subtree (and the full RJT) is the leftmost cluster. Finally, it is required that $I(j) \subset C_j$ for all $j \in N$. These properties result in a convenient structure where for each pair of adjacent clusters $(C_i, C_j) \in A_{RJT}$, we have $C_j \setminus C_i = j$, and the joint probability distribution of all nodes in C_j can thus always be obtained from the probability distribution of C_i and $\mathbb{P}(X_j = s_j \mid X_{I(j)} = s_{I(j)})$. As will be described next, this allows the problem to be formulated as a mixed-integer programming (MIP) model, which allows for employing standard techniques widely available in off-the-shelf solvers. For more detailed description of the properties of gradual RJTs, we refer the reader to Parmentier et al. (2020) and Herrala et al. (2023).

Let us define the binary variable $z(s_j \mid s_{I(j)})$ that takes value 1 if $Z_j(s_{I(j)}) = s_j$, and 0 otherwise, for all $j \in N^D$, $s_j \in S_j$, and $s_{I(j)} \in S_{I(j)}$. These variables correspond to the indicator function (1), representing local decision strategies at each decision node $j \in N^D$. Additionally, we define variables $\mu_{C_j} \geq 0$ representing the joint probability distribution of nodes $i \in C_j$. The expected utility maximization problem corresponding to the gradual RJT can then be formulated as

$$\max \sum_{j \in N^V} \sum_{s_{C_j} \in S_{C_j}} \mu_{C_j}(s_{C_j}) u_{C_j}(s_{C_j}) \tag{2}$$

$$\text{s.t.} \sum_{s_{C_j} \in S_{C_j}} \mu_{C_j}(s_{C_j}) = 1, \ \forall j \in N \tag{3}$$

$$\sum_{\substack{s_{C_i} \in S_{C_i}, \\ s_{C_i \cap C_j} = s^*_{C_i \cap C_j}}} \mu_{C_i}(s_{C_i}) = \sum_{\substack{s_{C_j} \in S_{C_j}, \\ s_{C_i \cap C_j} = s^*_{C_i \cap C_j}}} \mu_{C_j}(s_{C_j}), \ \forall (C_i, C_j) \in A_{RJT}, s^*_{C_i \cap C_j} \in S_{C_i \cap C_j} \tag{4}$$

$$\mu_{C_j}(s_{C_j}) = \mu_{\overline{C}_j}(s_{\overline{C}_j}) p(s_j \mid s_{I(j)}), \ \forall j \in N^C \cup N^V, s_{C_j} \in S_{C_j} \tag{5}$$

$$\mu_{C_j}(s_{C_j}) = \mu_{\overline{C}_j}(s_{\overline{C}_j}) \delta(s_j \mid s_{I(j)}), \ \forall j \in N^D, s_{C_j} \in S_{C_j} \tag{6}$$

$$\mu_{C_j}(s_{C_j}) \geq 0, \ \forall j \in N, s_{C_j} \in S_{C_j} \tag{7}$$

$$\delta(s_j \mid s_{I(j)}) \in \{0, 1\}, \ \forall j \in N^D, s_j \in S_j, s_{I(j)} \in S_{I(j)}. \tag{8}$$

The objective function (2) is the expected utility associated with the strategy $Z \in \mathbb{Z}$ represented by the decision variables z. The first constraint states that probability distributions μ_{C_j} must sum to one, and constraint (4) enforces local consistency between adjacent clusters. Here, local consistency means that the distribution $\mu_{C_i \cap C_j}$ must be the same when obtained as a marginal distribution from C_i or C_j. Constraints (5) and (6) propagate the probability information in the junction tree. For notational brevity, we use $\overline{C}_j = C_j \setminus j$, for which $\mu_{\overline{C}_j}(s_{\overline{C}_j}) = \sum_{s_j} \mu_{C_j}(s_{C_j})$. It should be noted that while constraint (6) contains a product of two decision variables, the decision strategy variables z are binary, making (6) an indicator constraint. This allows one to linearize the product using methods discussed in, e.g., Mitra et al. (1994), and the problem can be considered an instance of mixed integer linear programming (MILP).

3.3. Conditionally observed information

As originally proposed, the approaches to formulate influence diagrams as MILP models can only be applied to problems with Type 1 endogenous uncertainty, i.e., decision-dependent probabilities. However, many MSSP problems involve conditionally observed information, often also referred to as conditional information revelation.

Consider again an example of inspecting the engine of a car, where the decision maker (DM) can pay an expert to reveal information about the uncertain state of the engine before deciding whether or not to perform maintenance. A key concept with conditionally observed

information is *distinguishability*. Suppose we represent the state of the engine as a chance node j. The outcome of this node is independent of the DM's choice to inspect the engine and is observed after inspection. If we have two different states that the engine can be observed to be, say s_j and s'_j, from the DM's perspective, there is no difference between s_j and s'_j if the engine is not inspected and the state is thus not observed. In this case, such states are *indistinguishable*.

In conditionally observed information, the two key elements are the decisions or random events that the observation is conditional on, and the condition that the observation depends on. We refer to these as the *distinguishability set* $T_{i,j} \subset N^C \cup N^D$ and the *distinguishability condition* $F_{i,j} : S_{T_{i,j}} \to \{0, 1\}$, respectively. Here, $i \in N^C \cup N^D$ denotes the conditionally observed node, usually a chance node, and $j \in N^D$ is the decision node where that information is available if the distinguishability condition is fulfilled. Note that both i and $T_{i,j}$ must be contained in the cluster C_j, which can be achieved by making them a part of the information set $I(j)$ when converting the ID to an RJT.

In some cases, the conditionally observed node could also be a decision node. Such cases might arise in the context of distributed decision-making, where the decisions are made by multiple agents and observing the decision of another agent does not happen automatically, or problems in which previous decisions are not remembered by default and the decision maker must instead pay a price to retrieve information on past decisions. Distributed decision making in the context of influence diagrams is discussed in, e.g., Detwarasiti and Shachter (2005), and Piccione and Rubinstein (1997) point out that decision makers can often affect what they remember by choosing to keep track of information (including decisions) they would otherwise forget.

Using the notion of distinguishability sets and conditions, we can define *conditional arcs*

$$a_c \in A_c = \{(i, j) \mid i \in N^C \cup N^D, j \in N^D, i < j,\ T_{i,j} \neq \emptyset,\ F_{i,j}(s_{T_{i,j}}) = 1\} \tag{9}$$

to describe conditionally observed information in influence diagrams. Specifically, we say that a conditional arc a_c from node $i \in N^C \cup N^D$ to node $j \in N^D$ is active (i.e., node i is observed when making the decision corresponding to node j) if the condition $F_{i,j}$ is fulfilled by the states of the nodes $k \in N^C \cup N^D$ in the distinguishability set $T_{i,j}$. If $T_{i,j}$ is empty, there is no conditional observation of i in j and, thus, no conditional arc between these nodes exists. The concept is illustrated in Fig. 3.

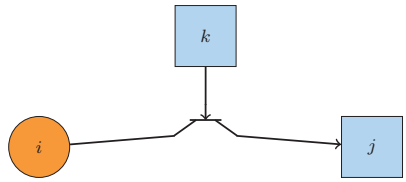


Figure 3: An illustration of a conditional arc (i, j). Nodes i and j correspond to the earlier notation where the realization of i is conditionally observed in j. The distinguishability set is $T_{i,j} = \{k\}$ and the middle part of the arc represents that the flow of information from i to j is conditional on k.

Using the example of inspecting a car engine before making a maintenance decision, the chance node i in Fig. 3 corresponds to the state of the engine, decision node k to the inspection decision, and j to a maintenance decision. The information about the engine is available when making the maintenance decision only if the decision maker chooses to inspect the engine. The distinguishability set is $T_{i,j} = \{k\}$, and the distinguishability condition is $F_{i,j}(s_{T_{i,j}}) = \mathbb{I}(s_k =$ "inspect engine"), where the indicator function $\mathbb{I}(\cdot)$ is defined as

$$\mathbb{I}(x = x^*) = \begin{cases} 1\,(true), & \text{if } x = x^*, \\ 0\,(false), & \text{otherwise.} \end{cases}$$

If the distinguishability set includes more than one node, alternative functions $F_{i,j}$ might be employed for modeling the conditional dependencies between the nodes. For example, if there are several projects that reveal the same information in node $i \in N$ and completing any of these projects is sufficient for information revelation, $F_{i,j}(s_{T_{i,j}}) = \bigvee_{k \in T_{i,j}} \mathbb{I}(s_k = s_k^*)$ can be used; or if all of the projects are required for the information revelation, $F_{i,j}(s_{T_{i,j}}) = \bigwedge_{k \in T_{i,j}} \mathbb{I}(s_k = s_k^*)$ is appropriate. An example of such conditions is found in Tarhan et al. (2009), where different uncertainties in oil field development are gradually revealed, and the uncertainty in the amount of recoverable oil in a reservoir can be resolved in two different ways, namely drilling a sufficient number of wells or using the reservoir for production for long enough.

3.4. Incorporating conditionally observed information in rooted junction trees

The conditional arcs are designed to describe conditionally observed information in influence diagrams. However, they are a general representation of the concept, not a modeling solution. In what follows, we present two alternative approaches for incorporating this concept into the RJT models, which ultimately enables solving Type 3 endogenously uncertain stochastic problems. The first approach employs *observation nodes*, used in decision analysis problems such as the used car buyer problem (Howard and Matheson, 2005) or the oil wildcatter problem (Raiffa, 1968). The second approach utilizes *conditional non-anticipativity constraints*, which are used in stochastic programming for modeling the decision-dependent information structure.

Observation nodes portray how the decision maker observes the information. By enforcing that earlier decisions affect the probability distribution of the observations, Type 2 uncertainty is effectively transformed into Type 1 uncertainty, making it directly amenable to an influence diagram. This approach is also used in Salo et al. (2022).

In effect, each conditional arc is replaced with an observation node, as illustrated in Fig. 4. The information set of the observation node is the union of the chance node i and the distinguishability set $T_{i,j}$ ($T_{i,j} = \{k\}$ in the example from Fig. 3), and the state space is $S_i \cup$ "no observation". Then, the observation node replaces the node $i \in N^C$ in the information set of $j \in N^D$, controlling whether or not the information in i is available in j. A benefit of this approach is that the modifications are done to the influence diagram, and the RJT formed from the resulting diagram can immediately be used in (2)-(8). Additionally, wrong or imperfect observations can also be modeled, e.g., if the inspector in the car engine example only provided an educated but nevertheless uncertain guess on the state of the engine.

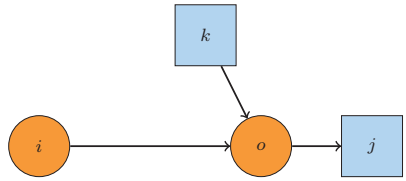


Figure 4: Replacing a distinguishability arc with an observation node in the example from Fig. 3.

One can also utilize the ideas of Ruszczyński (1997) and Apap and Grossmann (2017) for modeling conditional information revelation, in which the information structure can be connected to the decisions by using disjunctive constraints called conditional non-anticipativity constraints (C-NACs). These constraints are similar to the more traditional non-anticipativity constraints (Rockafellar and Wets, 1991) in stochastic programming, but the constraints are only imposed if the distinguishability conditions between the corresponding paths are satisfied.

To integrate C-NACs into our model, the conditional arcs introduced earlier need to be supplemented with the *conditional information set* $I_c(j) = \{i \in N^C \cup N^D \mid (i,j) \in A_c\}$ to represent the conditionally available information at node $j \in N^D$. The C-NACs then control whether or not this information is available when making the decision in node j. For the

upcoming developments, we require that $I_c(j) \subset C_j$ and $T_{i,j} \subset C_j$. We observe that two *cluster states* s_{C_j} and s'_{C_j} are always distinguishable at node $j \in N^D$ if the non-conditional information states $s_{I(j)}$ and $s'_{I(j)}$ are different. Instead, C-NACs are needed when the conditional information states differ between cluster states with the same non-conditional information state. For notational clarity, we separate the conditional and non-conditional information sets $I_c(j)$ and $I(j)$. C-NACs are used to enforce conditional non-anticipativity when only the conditional information states $s_{I_c(j)}$ differ between s_{C_j} and s'_{C_j}, and we use this notation to emphasize the fact that C-NACs are only introduced for such pairs.

If the conditional information states differ (i.e., node $i \in I_c(j)$ has different states in s_{C_j} and s'_{C_j}), distinguishability is dependent on the corresponding distinguishability condition(s) $F_{i,j}(s_{T_{i,j}})$. This distinguishability of two cluster states at node $j \in N^D$ can be formulated as a Boolean variable $f_j^{s_{C_j}, s'_{C_j}}$, defined as

$$f_j^{s_{C_j}, s'_{C_j}} = \bigvee_{i \in I_c(j) | s_i \neq s'_i} F_{i,j}(s_{T_{i,j}}).$$

The value of $f_j^{s_{C_j}, s'_{C_j}}$ is *True* (i.e., 1) if the conditionally revealed information makes scenarios s_{C_j} and s'_{C_j} distinguishable at node j, and *False* (i.e., 0) otherwise.

Finally, we extend the definition of the local decision strategy $Z_j(s_{I(j)})$ and the corresponding binary variables $z(s_j \mid s_{I(j)})$ to include the conditional information set $I_c(j)$. If the value of $f_j^{s_{C_j}, s'_{C_j}}$ is *False*, the local strategies $Z_j(s_{I(j)}, s_{I_c(j)})$ and $Z_j(s_{I(j)}, s'_{I_c(j)})$ must be the same. Combining these ideas, we can define C-NACs in the context of our model as

$$\neg f_j^{s_{C_j}, s'_{C_j}} \implies z(s_j \mid s_{I(j)}, s_{I_c(j)}) = z(s_j \mid s_{I(j)}, s'_{I_c(j)}), \forall s_{C_j}, s'_{C_j} \in S_{C_j}, . \tag{10}$$

In light of the above, the C-NACs for binary variables z can also be conveniently written as

$$|z(s_j \mid s_{I(j)}, s_{I_c(j)}) - z(s_j \mid s_{I(j)}, s'_{I_c(j)})| \leq f_j^{s_{C_j}, s'_{C_j}}, \forall s_{C_j}, s'_{C_j} \in S_{C_j}. \tag{11}$$

Notice that the absolute value function used in the left-hand side of (11) can be trivially linearized without significantly increasing the model complexity. This constraint states that for each decision node $j \in N^D$, if the non-conditional information states $s_{I(j)}$ are the same for two cluster states s_{C_j} and s'_{C_j}, and conditionally revealed information does not make these cluster states distinguishable either, the local decision strategy represented by the z-variables must be the same between them.

In practice, the main challenge with using C-NACs is that the number of constraints (11) quickly becomes overwhelmingly large. With this in mind, Apap and Grossmann (2017) present a number of C-NAC reduction properties that can be exploited to decrease the number of such constraints. By making use of these C-NAC reduction properties, representing the decision-dependent information structure within the RJT model might be more compact with C-NACs than the corresponding model using observation nodes. However, C-NACs lack the ability to model Type 1 endogenous uncertainty. Observation nodes are more versatile, allowing for modeling both types of endogenous uncertainty within one observation node, making it possible to model and solve T3ESP problems. In Section 4, we present a large-scale example problem involving different endogenous uncertainties.

4. Cost-benefit analysis for climate change mitigation

4.1. Model description

To illustrate the setting, we now consider the cost-benefit analysis on mitigating climate change under uncertainty (see, e.g., Ekholm, 2018). Climate change is driven by greenhouse

gas (GHG) emissions and can be mitigated by reducing these emissions, which incurs costs. However, mitigation reduces the negative impacts of climate change, referred to as climate damage. In cost-benefit analysis, the objective is to minimize the discounted sum of mitigation costs and climate damage over a long time horizon. However, multiple uncertainties complicate the analysis.

Here, we consider three salient uncertainties involving both decision-dependent probabilities (Type 1) and conditionally observed information (Type 2). Moreover, some decision nodes involve continuous variables. The resulting problem is a multi-stage mixed-integer nonlinear problem (MINLP) with Type 3 endogenous uncertainty, thus demonstrating the proposed novel features to the rooted junction tree framework described in this paper.

The considered uncertainties concern 1) the sensitivity of climate to GHG emissions, 2) the severity of climate change damages to society, and 3) the cost of reducing emissions. Decisions can be made to first conduct costly research and development (R&D) efforts towards each source of uncertainty. For the uncertainties regarding climate sensitivity and damages, a successful R&D effort results in an earlier revelation of the parametrization, whereas an unsuccessful or no R&D effort reveals this information later. Similar models of R&D pipeline optimization under (Type 2) endogenous uncertainty are considered in Colvin and Maravelias (2011).

For the mitigation costs, the model considers that technological R&D can be conducted to decrease the costs of bioenergy with carbon capture and storage (BECCS). These can take place at three intensity levels and in two distinct stages. The first stage is a choice between low or medium R&D effort. The low-effort choice represents a business-as-usual perspective, which carries throughout the decision process. If the medium effort is chosen, one observes whether the R&D looks promising or not, and can then decide whether to continue with the medium or switch to a higher R&D effort. The three R&D effort levels and whether the development seems promising or not all affect the probabilities for achieving either low, medium or high mitigation costs later during the century. For a related discussion, we refer the reader to Rathi and Zhang (2022), who consider endogenous technology learning on power generation.

The presented model is an extension from Ekholm and Baker (2022), which in turn is a simplification from the SCORE model (Ekholm, 2018). Compared to the formulation proposed here, these earlier analyses have assumed that the uncertainties are resolved exogenously over time and dealt with the mitigation cost uncertainty through separate scenarios. The details of the model structure and parametrization are described in Appendix B.

The influence diagram for the problem is presented in Fig. 5, and a corresponding RJT in Fig. 6. For converting the influence diagram into a rooted junction tree, we use the algorithm from Parmentier et al. (2020). The algorithm takes a set of nodes and their information sets, along with a topological order for the nodes and returns a gradual rooted junction tree. A topological order for a graph assigns a unique index to each node, so that for each arc $(i, j) \in A_{ID}$, the index of node j is larger than that of node i.

The first stage of the diagram involves R&D decisions towards climate sensitivity (D_{CS}), damages (D_{Dmg}) and technology (D_{T1}). If successful, the climate parameter (climate sensitivity and damage exponent) R&D efforts modify the information structure so that the parametrization is partially revealed in 2050 instead of 2070. This is represented by the nodes O_{Dmg} and O_{CS} and the outcome (success/failure) of the projects by C_{Dmg} and C_{CS}. Because the value nodes $v \in N^V$ represent deterministic mappings $s_{I(v)} \to \mathbb{R}$, the value nodes are not explicitly represented in the RJT. Instead, the components of the expected utility can be extracted from the clusters containing $I(v)$, following the ideas used in the computational experiments of Parmentier et al. (2020).

Decisions over emission reductions ($D_{Ei}, i \in \{1, 2, 3\}$) are made in three stages: in 2030, 2050 and 2070, which represent the medium-term and long-term climate actions. The technological R&D potentially lowers the costs of emission reductions in 2050 and 2070. We connect our example to the discussion on the feasibility of large-scale deployment of BECCS, which has

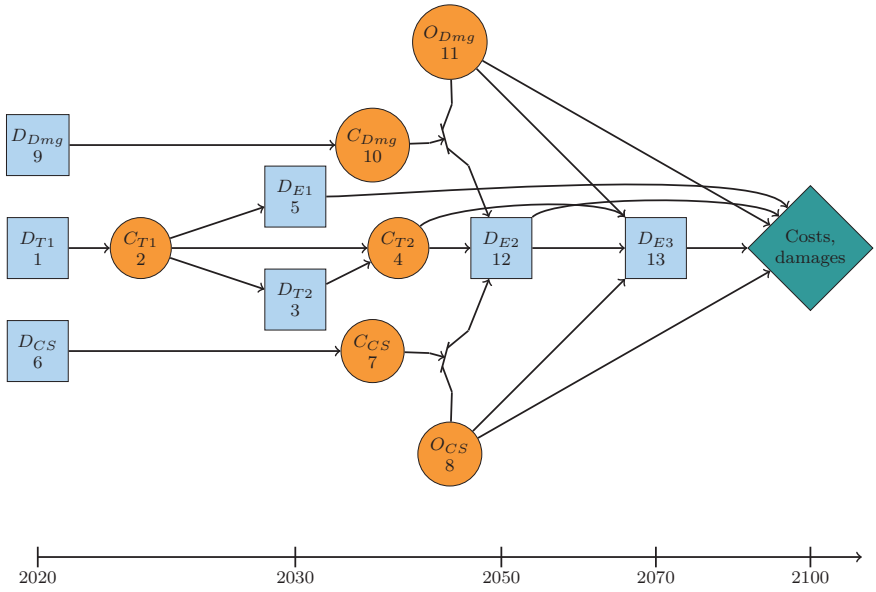


Figure 5: Influence diagram of the climate change cost-benefit problem with endogenous uncertainty due to R&D. We assume that all prior decisions and uncertainty realizations apart from the conditionally observed parameters are remembered when making decisions, but omit the arcs for clarity. Additionally, the individual value nodes associated with decision nodes 1, 3, 6 and 9 are omitted for clarity. The nodes are numbered according to the topological order used for forming the RJT.

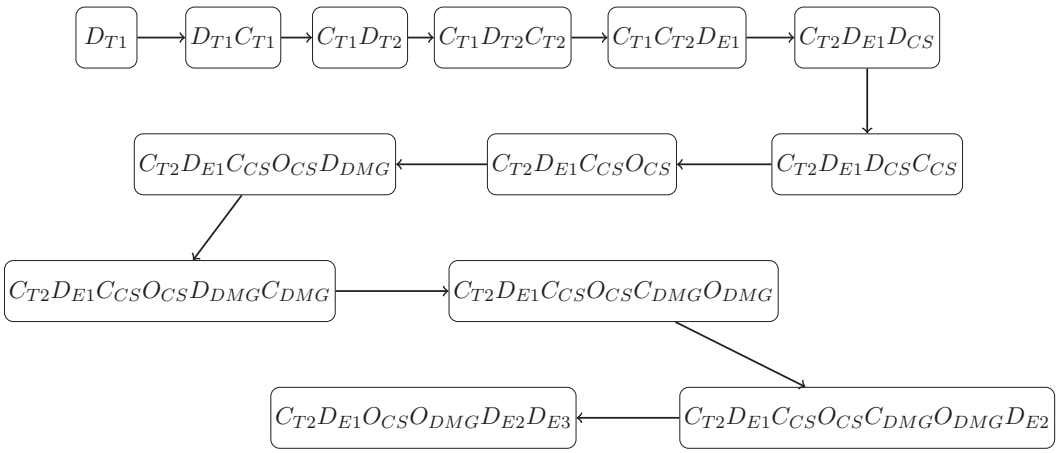


Figure 6: A rooted junction tree representation of the ID in Fig. 5

been a crucial but contested result of many mitigation scenarios (Calvin et al., 2021). The R&D costs and probabilities for the three levels of BECCS costs are parametrized using expert-elicited estimates in Baker et al. (2015). These are then reflected in the overall emission reduction costs. It is worth noting the major challenges in long-term technological foresight, which is manifested in the wide spectrum of responses from the experts; but elicitation data is nevertheless useful for illustrating the importance of technological progress.

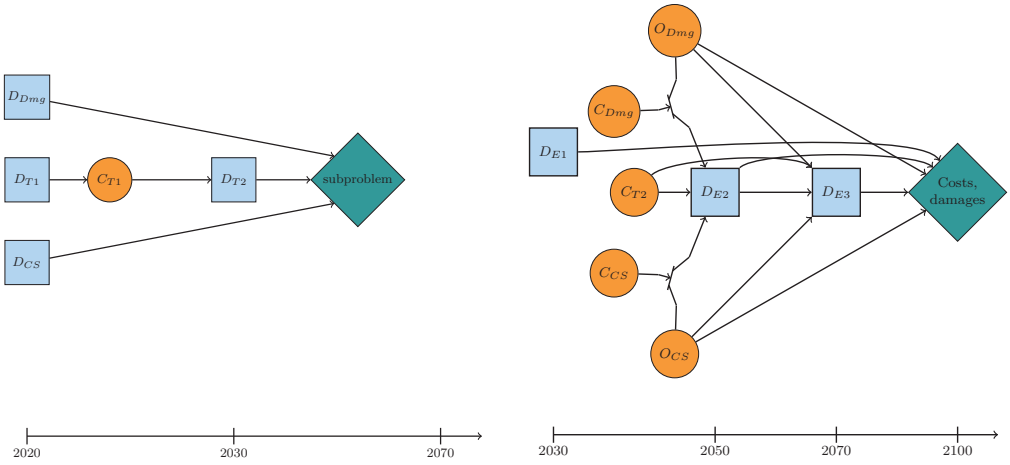
After 2070, the level of climate change is observed based on the chosen emission reductions

and the observed branch of climate sensitivity, which then determines the climate damages along with the observed branch of climate damages.

4.2. Modifying the influence diagram

As discussed in Section 3, the original formulation is limited to problems with discrete and finite state spaces for all nodes. However, discretizing the emission levels $D_{Ei}, i \in \{1, 2, 3\}$ would inevitably result in suboptimal solutions, thereby limiting the representability of the naturally continuous nature of the decision variables representing the emission levels.

To achieve this, we observe that the utilities in the value node(s) can be thought of as representing solutions to optimization problems. We thus modify the influence diagram and move the nodes $D_{Ei}, i \in \{1, 2, 3\}$ to a subproblem whose solution becomes the utility value in the main problem, loosely following the ideas in Lee et al. (2021). The influence diagram resulting from this decomposition is presented in Fig. 7.



(a) Influence diagram of the climate change cost-benefit main problem.

(b) Influence diagram of the climate change cost-benefit subproblem.

Figure 7: The decomposed climate change cost-benefit problem.

As a result, there is no Type 1 endogenous uncertainty in the subproblem and it can be modeled as a nonlinear three-stage stochastic programming problem with continuous decision variables. The nonlinearity comes from the SCORE model described in Appendix B. The impact of the decomposition on computational performance is discussed in Section 4.4.

4.3. Model results

The optimal R&D strategy for this problem is to carry out all R&D projects. Fig. 8 presents the emission levels of the optimal mitigation strategy. The effect of the technology R&D on optimal emission pathways is shown with the three subfigures corresponding to the final R&D outcome after 2030. Intuitively, successful research and therefore cheaper abatement leads to more abatement. This has also a major impact on the total costs of the optimal strategy: with the low cost curves, the total expected cost is roughly 30% lower than with medium costs, and with high abatement costs 30% higher than with the medium cost curve.

The climatic parameter R&D, the other endogenous effect in this model has a magnitude smaller effect on the expected costs than the technology R&D, but the effect on abatement levels is remarkable. The branches after 2030 represent different realizations of the technology research and partial learning of the climate parameters. If the research efforts for both parameters fail (blue lines in Fig. 8), the 2050 abatement decisions are made knowing only the outcome

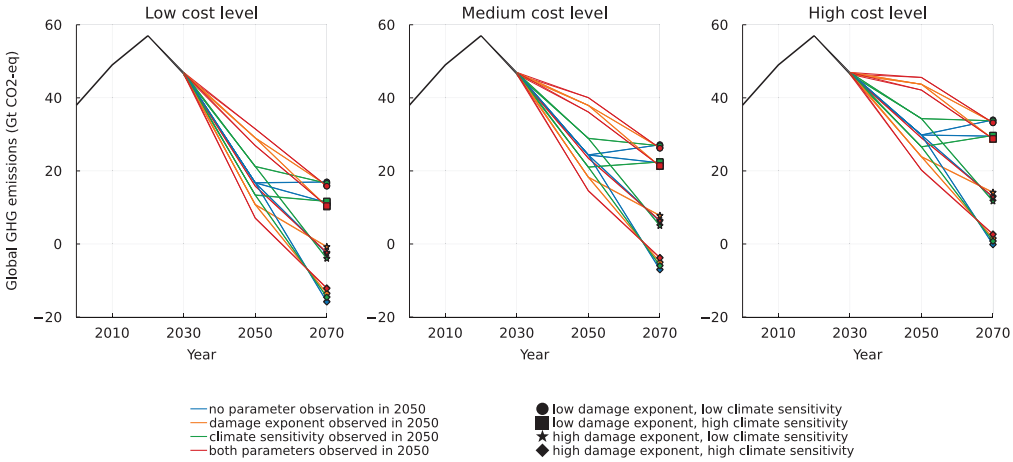


Figure 8: Abatement levels for different outcomes of technology research and parameter observations in 2030.

of the technology projects, and the four scenarios of partial learning only occur after 2050. Learning the parameters before 2050 results in more dispersed abatement strategies. Finally, the underlying parameter branching is represented with shapes in 2070. It can be seen that the impact of climate sensitivity is considerably smaller than that of the damage exponent. This is in line with the results in Ekholm and Baker (2022) and shows that the proposed framework could be applied in planning for optimal R&D pathways.

4.4. Computational aspects

Using the decomposed model presented in Fig. 7, the main problem is solved in 0.4 seconds, while solving all of the subproblems takes one second. If we instead solve a problem corresponding to Fig. 5 without decomposition, the framework requires discretizing the abatement levels. Using discrete decision spaces to approximate a continuous variable leads to suboptimal solutions. Furthermore, even with only five levels for each abatement decision, the problem becomes more computationally demanding than the continuous version. The damage costs are calculated in a tenth of a second, but solving the model takes four seconds. In larger problems, the discretized problem could quickly become intractable. The decomposed problem instead uses continuous decision variables to represent the abatement decisions, precluding the need for a discretization. As a consequence, it is faster to solve than the rudimentary approximation with five abatement levels.

The discretized model has 2172 constraints and 4332 variables, of which 327 are binary; and the main problem of the decomposed model has 40 constraints and 66 variables, of which 12 are binary. The discretized model is thus two orders of magnitude larger than the decomposed model, but the damage costs in the discretized model are calculated almost instantaneously. However, the trade-off of moving some of the computational burden into the subproblems makes it possible to solve the continuous problem to optimality. The discretized model can technically also be solved to optimality, but it requires a very rough discretization of the abatement levels, which are continuous decisions by nature. Even with a low number of abatement options, the solution times are larger than those for the decomposed model. In addition, such a rough discretization is likely to result in solutions that are far from the optimal values. While discretizing the decision spaces may seem like an obvious approach, as it enables direct implementation using the MIP formulation (2)-(8), it is not computationally viable in all cases. Overall, this case study provides an illustrative example of an application where the developments in this paper

make it possible to consider settings that are challenging both from modelling and computational standpoints.

5. Conclusion

In this paper, we propose a framework for Type 3 endogenously uncertain stochastic programming (T3ESP) problems. Our contributions consist of two different modelling approaches to consider conditionally observed information, making the framework more generally applicable to decision-making problems in contexts such as capacity expansion. The proposed framework is based on the work by Parmentier et al. (2020), originally developed for solving decision problems with decision-dependent uncertainties by converting an influence diagram representation of the problem to a mixed-integer programming (MIP) problem.

To the best of our knowledge, and in line with Hellemo et al. (2018), Type 3 endogenously uncertain stochastic programming (T3ESP) problems have not been previously addressed in the literature. To make RJT a framework applicable to such problems, we show how Type 2 endogenous uncertainties can be modeled by either adding observation nodes to the influence diagram or by adding conditional non-anticipativity constraints (C-NACs) to the model. We note that with minimal modifications, the developments presented in this paper can also be applied to extend the Decision Programming framework (Salo et al., 2022; Hankimaa et al., 2023).

In practice, both approaches have their advantages and disadvantages. Adding observation nodes only requires modifying the influence diagram by adding new nodes, while C-NACs are additional constraints that must be added to the decision model. On one hand, it would be beneficial to not require explicit modification of the decision model, i.e., the MIP model. On the other hand, if one allows for modifying the decision problem by using C-NACs, this might also allow for representing other parts of the decision process explicitly as added variables or constraints. Further research in this direction is thus relevant.

However, if decision-dependent probability distributions and conditional information revelation are intertwined in the problem structure, for example, by the presence of imperfect conditional observations, one cannot employ C-NAC constraints. An example of such a problem would be a version of the climate CBA problem in Section 4 where the climate parameter research does not reveal the correct branch, that is, remove one of the extreme parameter values, but instead gives a probability distribution that provides better information than the original. However, observation nodes can be applied even for such uncertainties.

Considering alternative modeling paradigms would make the framework suitable to a broader set of problems. Interesting examples of such research ideas are (distributionally) robust optimization and further examination of multi-objective decision-making. For large problems, it might be necessary to improve the computational performance using, e.g., decomposition methods for solving the MIP formulation (2)-(8), and solution heuristics. Influence diagram decomposition (e.g., Lee et al., 2021) can be used for improving the computational efficiency of finding maximum expected utility strategies for influence diagrams, and in the climate CBA problem, we demonstrate a simple influence diagram decomposition. Perhaps more interestingly, in the context of our MILP reformulations, we also show that such decomposition approaches can allow for solving decision problems with continuous decision variables, significantly improving the general applicability of the formulation.

In conclusion, the proposed developments turned the framework sufficiently general to model a challenging example problem with Type 3 endogenous uncertainty. It should also be noted that the influence diagrams are reformulated as MILP problems, guaranteeing global optimality of solutions despite the challenging nature of the underlying decision problems.

Acknowledgements

Funding: The work of Herrala and Oliveira was supported by the Research Council of Finland [*Decision Programming: A Stochastic Optimization Framework for Multi-Stage Decision Problems*, grant number 332180]; the work of Ekholm was supported by the Research Council of Finland [*Designing robust climate strategies with Negative Emission Technologies under deep uncertainties and risk accumulation (NETS)*, grant number 331764].

Appendix A. Computational environment

All problems are solved using an Intel E5-2680 CPU at 2.5GHz and 128GB of RAM, provided by the Aalto University School of Science Science-IT project. The problem code was implemented in Julia v1.7.3 (Bezanson et al., 2017) with the Gurobi solver v10.0.0 (Gurobi Optimization, LLC, 2022) and JuMP v1.5.0 (Dunning et al., 2017). All the code used in the computational experiments is available at a GitHub repository (Herrala, 2023).

Appendix B. Cost-benefit model description

The idea in climate change cost-benefit analysis (CBA) is the minimization of emission reduction costs and climate damages. The abatement cost calculation is based on marginal abatement cost curves, as presented in equation (B.1), using numerical estimates from the SCORE model (Ekholm, 2018). Coefficients α and β are the parameters of the cost curves, R is the total abatement level and c is the marginal cost of abatement. In (B.3), C is the total cost for abatement level R. The subscript t has been omitted for clarity, but parameters α and β change between stages due to assumed technological progress.

$$R = \alpha c^{\beta} \tag{B.1}$$

$$\implies c = \left(\frac{R}{\alpha}\right)^{1/\beta} \tag{B.2}$$

$$\implies C = \int_0^R \left(\frac{r}{\alpha}\right)^{1/\beta} dr = \frac{\beta}{1+\beta}\left(\frac{1}{\alpha}\right)^{1/\beta} R^{1+1/\beta}. \tag{B.3}$$

Departing from the predetermined technological progress that was assumed by Ekholm (2018), the parameter α in the model depends here on the result of technological R&D, as presented in Figure 5. We consider three levels of R&D effort in 2020 and 2030, which can then lead to three possible levels of MAC curves for years 2050 and 2070.

For the effect of R&D efforts on bioenergy and carbon capture and storage (CCS) costs, we used expert elicited estimates from Baker et al. (2015). To convert these into emission reduction costs, we assume a coal power plant as a baseline and calculate the additional costs from bioenergy with carbon capture and storage (BECCS) relative to the amount of reduced emissions by switching from coal to BECCS. Both plants were assumed to have a lifetime of 30 years and operate at 80% capacity on average. The coal power plant was assumed to have a 40% efficiency and produce 885 tonnes of CO_2 per GWh of electricity. The generation cost was assumed to be 50$/MWh. Costs were discounted at 5% rate.

Compared to coal, BECCS accrues additional costs per generated unit of electricity from the higher cost of biofuel and lower efficiency, and the additional investment to CCS and loss of efficiency from using some of the generated electricity in the carbon capture process. Baker et al. (2015) presented probability distributions for these parameters following three different levels of R&D efforts (low, medium and high). To calculate the cost differential to coal power plant, we performed a Monte Carlo sampling of these four parameters, separately for each R&D level,

which was then compared to the amount of reduced emissions per generated unit of electricity. This yields a distribution of emission reduction costs for BECCS for each R&D level.

We generalize the impact of R&D on BECCS's emission reduction cost to the overall marginal abatement cost (MAC) curve. This is obviously a simplification, but nevertheless reflects the major role that BECCS might have in decarbonizing the economy (e.g. Fuss et al., 2018; Rogelj et al., 2018). We take the high MAC from Ekholm (2018) as the starting point and define two MAC curves that are proportionally scaled down from the high MAC.

The low R&D level yields an average emission reduction cost of around 100 \$/t. We set three bins for three cost levels: high costs correspond to above 75 \$/t, medium costs are between 25 and 75 \$/t, and low costs are below 25 \$/t. The probabilities of achieving high, medium or low emission reduction costs are then estimated from the Monte Carlo sampling for each R&D level, and presented in Table B.1. With medium and high R&D effort, the average cost in the low cost bin is around 20 \$/t. Therefore we assign the reduction in the MAC as 50% for medium costs and 80% for low costs. The corresponding parameters are listed in Table B.2. These are still within the range of costs used in Ekholm (2018), where the low-cost MAC yielded the same emission reductions than the high-cost MAC with approximately 90% lower costs in 2050.

Table B.1: Probabilities for different abatement costs (rows) with different levels of R&D effort (columns).

	Low R&D	Medium R&D	High R&D
High costs	73 %	31 %	9 %
Medium costs	27 %	64 %	73 %
Low costs	0 %	5 %	18 %

Table B.2: Cost curve parametrization

year	α_{high}	α_{medium}	α_{low}	β
2030	3.57	3.57	3.57	0.340
2050	11.2	13.3	16.7	0.250
2070	21.1	24.3	29.3	0.203

The climate damage cost calculation is from DICE (Nordhaus, 2017). The damage function is presented in (B.4), where $Y(t)$ is the world gross economic output at time t, a is a scaling parameter and b is the damage exponent. While climate change and the abatement decisions have an effect on the economic output, the effect is assumed small and $Y(t)$ is defined exogenously in SCORE.

$$D(t, \Delta T) = Y(t) a \Delta T^b. \tag{B.4}$$

Finally, the temperature change ΔT is approximated with (B.5), where c is the climate sensitivity (the temperature increase from doubling of CO_2 emissions), M is the sum of emissions in 2030-2070 and k_i are coefficients.

$$\Delta T = k_1 c M + k_2 c + k_3 M + k_4. \tag{B.5}$$

In SCORE, both the DICE damage parameter and the climate sensitivity are uncertain with three options, low, medium and high, as presented in Table B.3, and the uncertainty is revealed in two steps in a binomial lattice. First, between 2050 and 2070, one of the extreme alternatives is removed from both uncertainties, that is, for both parameters, we know either that the value is not high or that it is not low. Then, after 2070, we learn the actual value.

The implementation here combines influence diagrams and MSSP in a way that the underlying branching probabilities are used as the probabilities of the observations O_{Dmg} and O_{CS} in Fig. 5. For the damage exponent, all branching probabilities are 50%. The observation O_{Dmg}

thus has a 50% probability of removing either the high or low value. Depending on the branch, the low or high value then has a 50% probability in the later branching, with the other 50% for the medium value. This makes the medium branch have a 50% probability in total, while the two extreme values both have a 25% probability. For the climate sensitivity, the first branching is with a 50% probability for both branches. However, the second branching is different. If the high sensitivity is excluded in the first branch, there is a 21% conditional probability of the low branch in the second branching, meaning a 10.5% total probability for the low sensitivity. Similarly, there is a 23% conditional probability of high damages in the other branch, resulting in a 11.5% probability for the high sensitivity. The remaining 78% is the final probability of medium sensitivity.

Table B.3: Climate sensitivity and damage exponent values

	Climate sensitivity	Damage exponent
High	6	4
Medium	3	2
Low	1.5	1

This uncertain process is modeled by means of a multi-stage stochastic programming problem, where new information is obtained gradually. It is possible to perform research on these parameters. If the research succeeds, one of the extreme values is excluded already before 2050, revealing the first branching in the observation process earlier than without or with failed research. The observation of the actual parameter value (the second branching) still happens after 2070, after all abatement decisions have been made. The total cost we aim to minimize is then a discounted sum of research costs, abatement costs (B.3) for years 2030, 2050 and 2070, and damage costs (B.4).

References

Apap, R.M., Grossmann, I.E., 2017. Models and computational strategies for multistage stochastic programming under endogenous and exogenous uncertainties. Computers & Chemical Engineering 103, 233–274. https://doi.org/10.1016/j.compchemeng.2016.11.011.

Baker, E., Bosetti, V., Anadon, L.D., Henrion, M., Aleluia Reis, L., 2015. Future costs of key low-carbon energy technologies: Harmonization and aggregation of energy technology expert elicitation data. Energy Policy 80, 219–232. https://doi.org/10.1016/j.enpol.2014.10.008.

Bezanson, J., Edelman, A., Karpinski, S., Shah, V.B., 2017. Julia: A fresh approach to numerical computing. SIAM Review 59, 65–98. https://doi.org/10.1137/141000671.

Bielza, C., Gómez, M., Shenoy, P.P., 2011. A review of representation issues and modeling challenges with influence diagrams. Omega 39, 227–241.

Boland, N., Dumitrescu, I., Froyland, G., 2008. A multistage stochastic programming approach to open pit mine production scheduling with uncertain geology. Optimization online , 1–33.

Calvin, K., Cowie, A., Berndes, G., Arneth, A., Cherubini, F., Portugal-Pereira, J., Grassi, G., House, J., Johnson, F.X., Popp, A., Rounsevell, M., Slade, R., Smith, P., 2021. Bioenergy for climate change mitigation: scale and sustainability. GCB Bioenergy 13, 1346–1371. https://doi.org/10.1111/gcbb.12863.

Colvin, M., Maravelias, C.T., 2010. Modeling methods and a branch and cut algorithm for pharmaceutical clinical trial planning using stochastic programming. European Journal of Operational Research 203, 205–215. https://doi.org/10.1016/j.ejor.2009.07.022.

Colvin, M., Maravelias, C.T., 2011. R&d pipeline management: Task interdependencies and risk management. European Journal of Operational Research 215, 616–628. https://doi.org/10.1016/j.ejor.2011.06.023.

Detwarasiti, A., Shachter, R.D., 2005. Influence diagrams for team decision analysis. Decision Analysis 2, 207–228.

Dunning, I., Huchette, J., Lubin, M., 2017. JuMP: A Modeling Language for Mathematical Optimization. SIAM Review 59, 295–320. https://doi.org/10.1137/15M1020575.

Dupačová, J., 2006. Optimization under exogenous and endogenous uncertainty, in: Proceedings of the 24th International conference on Mathematical Methods in Economics, pp. 131–136. https://doi.org/10.13140/2.1.2682.2089.

Ekholm, T., 2018. Climatic Cost-benefit Analysis Under Uncertainty and Learning on Climate Sensitivity and Damages. Ecological Economics 154, 99–106. https://doi.org/10.1016/j.ecolecon.2018.07.024.

Ekholm, T., Baker, E., 2022. Multiple Beliefs, Dominance and Dynamic Consistency. Management Science 68, 529–540. https://doi.org/10.1287/mnsc.2020.3908.

Escudero, L.F., Garín, M.A., Monge, J.F., Unzueta, A., 2020. Some matheuristic algorithms for multistage stochastic optimization models with endogenous uncertainty and risk management. European Journal of Operational Research 285, 988–1001. https://doi.org/10.1016/j.ejor.2020.02.046.

Fuss, S., Lamb, W.F., Callaghan, M.W., Hilaire, J., Creutzig, F., Amann, T., Beringer, T., De Oliveira Garcia, W., Hartmann, J., Khanna, T., Luderer, G., Nemet, G.F., Rogelj, J., Smith, P., Vicente, J.V., Wilcox, J., Del Mar Zamora Dominguez, M., Minx, J.C., 2018. Negative emissions - Part 2: Costs, potentials and side effects. Environmental Research Letters 13. https://doi.org/10.1088/1748-9326/aabf9f.

Goel, V., Grossmann, I.E., 2006. A Class of stochastic programs with decision dependent uncertainty. Mathematical Programming 108, 355–394. https://doi.org/10.1007/s10107-006-0715-7.

Gurobi Optimization, LLC, 2022. Gurobi Optimizer Reference Manual. URL: `https://www.gurobi.com`.

Gustafsson, J., Salo, A., 2005. Contingent portfolio programming for the management of risky projects. Operations research 53, 946–956. https://doi.org/10.1287/opre.1050.0225.

Hankimaa, H., Herrala, O., Oliveira, F., Tollander de Balsch, J., 2023. Decisionprogramming.jl – a framework for modelling decision problems using mathematical programming. `arXiv:2307.13299`.

Hellemo, L., Barton, P.I., Tomasgard, A., 2018. Decision-dependent probabilities in stochastic programs with recourse. Computational Management Science 15, 369–395. https://doi.org/10.1007/s10287-018-0330-0.

Herrala, O., 2023. Source code repository: Type 2 RJT. URL: `https://github.com/solliolli/type2-rjt`.

Herrala, O., Terho, T., Oliveira, F., 2023. Risk measures in rooted junction tree models. In preparation .

Howard, R.A., Matheson, J.E., 2005. Influence diagrams. Decision Analysis 2, 127–143. https://doi.org/10.1287/deca.1050.0020.

Jonsbråten, T.W., Wets, R.J., Woodruff, D.L., 1998. A class of stochastic programs with decision dependent random elements. Annals of Operations Research 82, 83–106. https://doi.org/10.1023/A:1018943626786.

Lauritzen, S.L., Nilsson, D., 2001. Representing and solving decision problems with limited information. Management Science 47, 1235–1251. https://doi.org/10.1287/mnsc.47.9.1235.9779.

Lee, J., Marinescu, R., Dechter, R., 2021. Submodel decomposition bounds for influence diagrams, in: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 12147–12157.

Li, X., Liu, Q., 2023. Strategic ignorance: Managing endogenous demand in a supply chain. Omega 114, 102729. https://doi.org/10.1016/j.omega.2022.102729.

Mitra, G., Lucas, C., Moody, S., Hadjiconstantinou, E., 1994. Tools for reformulating logical forms into zero-one mixed integer programs. European Journal of Operational Research 72, 262–276.

Nordhaus, W.D., 2017. Revisiting the social cost of carbon. Proceedings of the National Academy of Sciences 114, 1518–1523. https://doi.org/10.1073/pnas.1609244114.

Ntaimo, L., Arrubla, J.A.G., Stripling, C., Young, J., Spencer, T., 2012. A stochastic programming standard response model for wildfire initial attack planning. Canadian Journal of Forest Research 42, 987–1001. https://doi.org/10.1139/x2012-032.

Parmentier, A., Cohen, V., Leclère, V., Obozinski, G., Salmon, J., 2020. Integer programming on the junction tree polytope for influence diagrams. INFORMS Journal on Optimization 2, 209–228.

Peeta, S., Sibel Salman, F., Gunnec, D., Viswanath, K., 2010. Pre-disaster investment decisions for strengthening a highway network. Computers and Operations Research 37, 1708–1719. https://doi.org/10.1016/j.cor.2009.12.006.

Piccione, M., Rubinstein, A., 1997. On the interpretation of decision problems with imperfect recall. Games and Economic Behavior 20, 3–24.

Raiffa, H., 1968. Decision analysis : introductory lectures on choices under uncertainty. Random House, New York.

Rathi, T., Zhang, Q., 2022. Capacity planning with uncertain endogenous technology learning. Computers & Chemical Engineering 164, 107868. https://doi.org/10.1016/j.compchemeng.2022.107868.

Rockafellar, R.T., Wets, R.J.B., 1991. Scenarios and policy aggregation in optimization under uncertainty. Mathematics of operations research 16, 119–147. https://doi.org/10.1287/moor.16.1.119.

Rogelj, J., Popp, A., Calvin, K.V., Luderer, G., Emmerling, J., Gernaat, D., Fujimori, S., Strefler, J., Hasegawa, T., Marangoni, G., Krey, V., Kriegler, E., Riahi, K., Van Vuuren, D.P., Doelman, J., Drouet, L., Edmonds, J., Fricko, O., Harmsen, M., Havlík, P., Humpenöder, F., Stehfest, E., Tavoni, M., 2018. Scenarios towards limiting global mean temperature increase below 1.5 C. Nature Climate Change 8, 325–332. https://doi.org/10.1038/s41558-018-0091-3.

Ruszczyński, A., 1997. Decomposition methods in stochastic programming. Mathematical Programming, Series B 79, 333–353. https://doi.org/10.1007/BF02614323.

Salo, A., Andelmin, J., Oliveira, F., 2022. Decision programming for mixed-integer multi-stage optimization under uncertainty. European Journal of Operational Research 299, 550–565. https://doi.org/10.1016/j.ejor.2021.12.013.

Solak, S., Clarke, J.P., Johnson, E., Barnes, E., 2008. A stochastic programming model with decision dependent uncertainty realizations for technology portfolio management, in: Operations Research Proceedings 2007, Springer. pp. 75–80. https://doi.org/10.1007/978-3-540-77903-2_12.

Tarhan, B., Grossmann, I.E., Goel, V., 2009. Stochastic programming approach for the planning of offshore oil or gas field infrastructure under decision-dependent uncertainty. Industrial and Engineering Chemistry Research 48, 3078–3097. https://doi.org/10.1021/ie8013549.

Zhou, R., Bhuiyan, T.H., Medal, H.R., Sherwin, M.D., Yang, D., 2022. A stochastic programming model with endogenous uncertainty for selecting supplier development programs to proactively mitigate supplier risk. Omega 107, 102542. https://doi.org/10.1016/j.omega.2021.102542.

Publication IV

Olli Herrala, Steven A. Gabriel, Fabricio Oliveira, Tommi Ekholm. A novel strong duality -based reformulation for trilevel infrastructure models in energy systems development. *Journal of the Operational Research Society*, July 2024.

JOURNAL OF THE OPERATIONAL RESEARCH SOCIETY
https://doi.org/10.1080/01605682.2024.2365807

ORIGINAL ARTICLE

OPEN ACCESS Check for updates

A novel strong duality-based reformulation for trilevel infrastructure models in energy systems development

Olli Herrala[a], Steven A. Gabriel[a,b,c], Fabricio Oliveira[a] and Tommi Ekholm[a,d]

[a]Aalto University, Espoo, Finland; [b]University of Maryland, College Park, Maryland, USA; [c]Norwegian University of Science and Technology, Trondheim, Norway; [d]Finnish Meteorological Institute, Helsinki, Finland

ABSTRACT

We explore the class of trilevel equilibrium problems with a focus on energy-environmental applications and present a novel single-level reformulation for such problems, based on strong duality. To the best of our knowledge, only one alternative single-level reformulation for trilevel problems exists. This reformulation uses a representation of the bottom-level solution set, whereas we propose a reformulation based on strong duality. Our novel reformulation is compared to this existing formulation, discussing both model sizes and computational performance. In particular, we apply this trilevel framework to a power market model, exploring the possibilities of an international policymaker in reducing emissions of the system. Using the proposed approach, we are able to obtain globally optimal solutions for a five-node case study representing the Nordic countries and assess the impact of a carbon tax on the electricity production portfolio.

ARTICLE HISTORY
Received 14 September 2023
Accepted 3 June 2024

KEYWORDS
Equilibrium modelling; bilevel optimisation; power systems modelling

1. Introduction

Hierarchical optimisation models with three levels of decision-makers arise in contexts such as traffic equilibrium (Gabriel, et al., 2022; Gu et al., 2019) and electricity market modelling (Huppmann and Egerer, 2015; Jin and Ryan, 2014). The hierarchical structure can be, e.g., such that the bottom-level players use a network operated by a middle-level player and regulated by a top-level player. For both electricity and traffic networks, similar models without the top-level regulators have been explored using *bilevel optimisation*, see Sinha et al. (2018) for a review.

Albeit challenging from both methodological and computational standpoints, including a top-level regulator as the third level, as opposed to considering only bilevel models, can provide important policy insights. In the particular case of energy systems, these models can yield more realistic solutions in which more stakeholders are assumed to act in coordination considering their own objectives. Obtaining equilibrium solutions for these models can thereby provide policy insights on pathways towards decarbonisation goals. Gabriel, et al. (2022) present a single-level reformulation for bilevel problems with complementarity-constrained bottom levels and discuss the possibility of using the model in a trilevel power market setting. However, their article includes no computational experiments demonstrating the practical usability of the proposed methods. Our aim is to explore the computational efficiency of the method using an illustrative power system setting representing the market structure in the Nordic countries.

The contribution of this article is twofold. First, in Section 2, we present background on bi- and trilevel optimisation, ending with our novel approach for solving trilevel equilibrium problems based on strong duality in Section 2.4. Compared to the formulation presented in Gabriel, et al. (2022) and summarised in Section 2.3, our proposed formulation results in fewer constraints, which is likely to result in increased computational efficiency. Second, we illustrate the methodological contributions using a stylised trilevel power market model described in Section 3. The motivation for our application stems from the recent discussion about optimal carbon taxation and its impact on electricity production (e.g., Hájek et al., 2019). The computational performance of the model is explored in Section 4.1, and finally, in Section 4.2, we apply the trilevel equilibrium modelling framework to a power market case study based on Belyak et al. (2023). These contributions are significant for the novel class of trilevel optimisation problems, and equilibrium modelling area in general. Finally, Section 5 concludes the article and discusses future research directions.

CONTACT Fabricio Oliveira fabricio.oliveira@aalto.fi Aalto University, Espoo, PO Box 11000 FI-00076 AALTO, Finland

2. Background

2.1. Earlier research

Bilevel optimisation considers problems with a hierarchical structure consisting of an upper-level player and one or more lower-level players (Bard, 1983). In power sector models (Gabriel, et al., 2012), the upper-level player is often a transmission system operator and the bottom level consists of electricity producers in a Cournot oligopoly. The general structure of a bilevel problem with linear upper- and lower-level problems is presented in (1) and (2). The upper-level problem P_u is

$$(P_u): \ min_{x,y\geq 0} \quad c_1^\top x + d_1^\top y \tag{1a}$$
$$\text{s.t.} \quad A_1 x + B_1 y \geq a_1, \tag{1b}$$
$$y \text{ solves } P_l(x), \tag{1c}$$

where $P_l(x)$ denotes the lower-level problem. Here, c_1, $x \in R^{n_x}$, d_1, $y \in R^{n_y}$, $A_1 \in R^{m_u \times n_x}$, $B_1 \in R^{m_u \times n_y}$, and $a_1 \in R^{m_u}$. The overall idea of this formulation is that the upper-level player's decision variable x affects the lower-level players' optimal decisions y, which is reflected back to the upper level in the constraint (1c). The linear lower-level problem is formulated as

$$(P_l(x)): \ min_{y\geq 0} \quad d_2^\top y \tag{2a}$$
$$\text{s.t.} \quad A_2 x + B_2 y \geq a_2. \tag{2b}$$

In general, both problems can also include equality constraints, but they have been omitted here for brevity, without loss of generality.

Solution methods for bilevel problems are based on the idea of replacing the upper-level constraint (1c) with the optimality conditions of the lower-level problem (2). The two main alternatives are the Karush–Kuhn–Tucker (KKT) optimality conditions (Karush, 1939; Kuhn and Tucker, 1951), leading to a mathematical program with equilibrium constraints (MPEC); and mathematical programming with primal and dual constraints (MPPDC) (Baringo and Conejo, 2012; Ruiz et al., 2012). Additionally, approaches based on optimal value functions (Ye and Zhu, 1995) can be used. Bilevel optimisation models can be used in contexts such as Stackelberg games (Bard, 1991), Cournot competition (Gabriel, et al., 2012), and robust optimisation (Leyffer et al., 2020). For a recent survey on applications and algorithms for bilevel optimisation, we refer to Kleinert et al. (2021).

A major challenge in bilevel optimisation is that, even for linear bilevel problems, single-level reformulations using complementarity constraints lead to non-linear and non-convex problems. This significantly increases the computational complexity of solving such problems and requires specialised approaches such as the simplex method-inspired projected gradient method by Still (2002) or the (spatial) branch-and-bound methods discussed by Bard and Moore (1990) and implemented in the Gurobi solver (Gurobi Optimization, LLC, 2022). Alternatively, one can use heuristics such as genetic programming (Kieffer et al., 2020) and particle swarm optimisation (Gao and Liu, 2021). For a review on heuristic solution methods for bilevel programming, we refer the reader to Camacho-Vallejo et al. (2023).

2.2. Trilevel equilibrium models

Consider a problem with a trilevel structure, in which players interact with each other at all three levels: top, middle, and bottom. In this structure, the top-level problem P_1 is assumed to be a linear optimisation problem with the middle-level problem $P_2(x)$ represented by the constraint (3c).

$$(P_1): \ min_{x,y,z} \quad c_1^\top x + d_1^\top y + e_1^\top z \tag{3a}$$
$$\text{s.t.} \quad A_1 x + B_1 y + C_1 z \geq a_1 \tag{3b}$$
$$y, z \text{ solve } P_2(x), \tag{3c}$$

where $P_2(x)$ denotes the middle-level problem

$$(P_2(x)): \ min_{y,z} \quad d_2^\top y + e_2^\top z \tag{4a}$$
$$\text{s.t.} \quad A_2 x + B_2 y + C_2 z \geq a_2 \quad (\gamma) \tag{4b}$$
$$y \geq 0 \tag{4c}$$
$$z \text{ solves } P_3(x,y), \tag{4d}$$

where γ is the vector of dual variables associated with constraint (4b) and z is the vector of bottom-level variables. In trilevel settings, the "lower-level" problem $P_2(x)$ is itself a bilevel problem. This is challenging, because bilevel optimisation problems are generally non-convex and directly obtaining their optimality conditions is thus difficult. The middle-level problem $P_2(x)$ constraints contain a bottom-level problem $P_3(x,y)$ that is parameterised by the upper-level variables x and middle-level variables y. In particular, Gabriel, et al. (2022) discuss the case of the bottom-level problem being a linear complementarity problem (LCP)

$$P_3(x,y): \quad 0 \leq \tilde{z} \perp q + N_x x + N_y y + M\tilde{z} \geq 0 \tag{5}$$

parameterised via the vector terms $N_x x$ and $N_y y$. It should be noted that (5) can be viewed as the KKT conditions of convex quadratic problems and in Section 2.4, we discuss such problems in more detail. Hereinafter, we use the standard $\perp$-notation

$$0 \leq a \perp b \geq 0 \iff a, b \geq 0, a^\top b = 0$$

for complementarity constraints with vectors a and b.

Trilevel problems have been researched by, e.g., Sauma and Oren (2007), who do not solve the models directly and instead iteratively solve the

middle- and bottom-level problems for different values of the top-level decision variables, and Dvorkin et al. (2018), who employ a column-and-constraint generation algorithm. In contrast, the goal of this article is to explore novel single-level reformulations for trilevel problems and solve them using an off-the-shelf solver, thus lowering the barrier-to-entry for using these models. However, this comes at the expense of imposing constraints on the structure of the models that are tractable in this manner. In (4), a general form of the problem is used, and the bottom-level problem in (4d) is assumed to be parameterised by both x and y. For the sake of clarity, we define two classes of trilevel problems with different degrees of computational challenges.

Definition 2.1. *If P_3 is parameterised by both x and y, we say that the problem has a strong trilevel structure. In contrast, if P_3 is parameterised only by the top-level variables x, i.e., it is not directly dependent on y, we say that the problem has a weak trilevel structure.*

Gabriel, et al. (2022) show that in order to use their single-level reformulation, the problem must have a weak trilevel structure, allowing such problems to be solved rather effectively by borrowing from the results in Cottle et al. (2009, Theorem 3.1.6) as long as the matrix M in the lower-level problem (5) is positive semi-definite (PSD). We also show that our novel reformulation in Section 2.4 retains this structural limitation and that the energy-environmental planning problem considered in this article has this structure. The aim of this article is to develop an alternative reformulation improving computational tractability and efficiency compared to the reformulation in Gabriel, et al. (2022), and the discussion on ways for lifting this restriction on problem structure is outside the scope of this article.

While the lack of direct influence for the middle-level player is a limitation, there are still structures that necessitate the use of a trilevel framework. As an example of a setting where a trilevel approach is required, we use the power market example in Section 3, where the bottom level consists of electricity generators, and on the middle level we have a profit-maximising system operator who has to satisfy a minimum renewable share in electricity production. If the bottom-level LCP matrix M in (5) is PSD, the bottom-level problem can have multiple optima. This could result in, e.g., a situation where it makes no difference for a generator to produce electricity using coal in one node or wind power in another.

Using the optimistic bilevel assumption (Dempe and Zemkoho, 2020), while the system operator cannot directly influence the generators, they can choose a bottom-level optimum that maximises their profit while satisfying the minimum renewable share constraint. In turn, maximising the middle-level player's profit could, in some settings, result in worse objective values for the top-level player. These interactions could not be represented in a setting where the middle-level player is insensitive to the bottom-level player's decisions, as the middle-level player must consider the bottom-level optimality conditions to be able to choose between bottom-level optimal solutions.

2.3. Bottom-level LCP with a positive semi-definite coefficient matrix

For completeness, we summarise the solution approach introduced in Gabriel, et al. (2022). Let us assume that the matrix M in (5) is PSD and that we have a solution $\bar{z}$ of (5). Furthermore, we assume the problem to have a weak trilevel structure and thus $N_y = 0$, i.e., the middle-level decisions y do not influence the bottom-level problem (c.f. Definition 2.1). Gabriel, et al. (2022) show that for a PSD M and a weak trilevel structure, a solution to the trilevel problem consisting of (3)–(5) can be obtained by solving the equivalent single-level reformulation

$$\min_{x,y,\tilde{z},\bar{z},\beta,\gamma,\delta,\zeta,\eta} \quad c_1^\top x + d_1^\top y + e_1^\top \tilde{z} \tag{6a}$$

$$\text{s.t.} \quad A_1 x + B_1 y + C_1 \tilde{z} \geq a_1, \tag{6b}$$

$$0 \leq y \perp d_2 - B_2^\top \gamma \geq 0, \tag{6c}$$

$$\begin{aligned} 0 \leq \tilde{z} \perp e_2 - C_2^\top \gamma - M^\top \delta - \zeta(q + N_x x) \\ - (M + M^\top)^\top \eta \geq 0, \end{aligned} \tag{6d}$$

$$0 \leq \bar{z} \perp q + N_x x + (M + M^\top)\bar{z} - M^\top \beta \geq 0, \tag{6e}$$

$$0 \leq \beta \perp q + N_x x + M\bar{z} \geq 0, \tag{6f}$$

$$0 \leq \delta \perp (q + N_x x) + M\tilde{z} \geq 0, \tag{6g}$$

$$(q + N_x x)^\top (\tilde{z} - \bar{z}) = 0, \;\; (M + M^\top)(\tilde{z} - \bar{z}) = 0, \tag{6h}$$

$$0 \leq \gamma \perp A_2 x + B_2 y + C_2 \tilde{z} - a_2 \geq 0, \tag{6i}$$

where $\bar{z}$ is a solution to the bottom-level problem (5) and $(\bar{z}^*)^\top (q + N_x x^* + M\bar{z}^*) = 0$ has to thus hold at an optimal solution $x^*, \bar{z}^*$ to (6). Appendix A summarises the reformulation steps taken in Gabriel, et al. (2022), including the constraints corresponding to the dual variables $\beta, \delta, \zeta,$ and η. This formulation assumes non-negativity for all variables y, but we note that this is not a requirement and including free variables in the middle level only requires small changes to the corresponding KKT conditions (6c).

Finally, we note that the reformulation (6) is not linear due to the non-linear products $\zeta N_x x$, $x^\top N_x^\top \tilde{z}$ and $x^\top N_x^\top \bar{z}$, resulting in a non-convex problem. In

general, obtaining global optimal solutions to non-convex problems is enormously challenging, but this particular non-convexity can be handled by using a solver capable of handling problems with bilinear terms in special ordered sets of type 1 (SOS1) constraints (Beale and Tomlin, 1970). An SOS1 constraint states that out of a set of variables or functions, only one can have a non-zero value. A complementarity constraint $0 \leq a \perp b \geq 0$ can thus be reformulated as two non-negative variables a and b in an SOS1 constraint. This can be achieved using, e.g., the spatial branch-and-bound method in the Gurobi solver (Gurobi Optimization, LLC, 2022); see also Siddiqui and Gabriel (2013).

2.4. Mathematical programming with complementarity from primal and dual constraints

We are now ready to discuss our novel single-level reformulation. Let us first consider a setting where the bottom level is a convex quadratic minimisation problem. So far, we have discussed a reformulation based on adding the KKT optimality conditions of the bottom-level problem to the middle-level problem. In our trilevel case, the KKT optimality conditions, having complementarity constraints, require a reformulation of the LCP solution set so that we can obtain a single-level equivalent formulation of the trilevel problem. This eventually results in the middle- and bottom-level problems being represented as two optimisation problems, potentially leading to computational challenges with the reformulation (6). Representing these two nested optimisation problems as a single-level equivalent requires a large number of complementarity constraints (6e)–(6i), possibly leading to prohibitive computational requirements.

To circumvent these challenges, we note that some bilevel optimisation problems can also be reformulated as MPPDC, using strong duality instead of complementarity. We present a novel strong duality-based reformulation for trilevel problems, in which a linear middle-level problem and convex quadratic bottom-level problems are reformulated into a single quadratically constrained linear problem instead of two optimisation problems (a quadratic program (QP) and a linear program (LP)) as in Appendix A and Gabriel, et al. (2022). The model sizes resulting from using complementarity (Section 2.3) and strong duality (this section) for the bottom level are compared in Section 2.5.

Consider a trilevel problem with a set of bottom-level problems $P_{3i}(x)$

$$(P_{3i}(x)): \quad \min_{z_i} \quad \frac{1}{2} z_i^\top F_i z_i + e_{i3}(x)^\top z_i \tag{7a}$$

$$\text{s.t.} \quad C_{i3} z_i \geq a_{i3}(x) \tag{7b}$$

$$z_i \geq 0, \tag{7c}$$

where z_i is a vector of decision variables and F_i is PSD for all $i \in I$. In our illustrative example described in the next section, the set I represents the electricity producers. Note that we assume a weak trilevel structure, that is, P_{3i} does not depend on y. Dorn (1960) presents Lagrangian dual formulations for quadratic problems,[1] and using these formulations, the dual of each problem $P_{3i}(x)$ is

$$\max_{p_i, z_i} \quad -\frac{1}{2} z_i^\top F_i z_i + a_{i3}(x)^\top p_i \tag{8a}$$

$$\text{s.t.} \quad C_{i3}^\top p_i - F_i z_i \leq e_{i3}(x) \tag{8b}$$

$$p_i \geq 0. \tag{8c}$$

In MPPDC, the complementarity constraints in the KKT optimality conditions are replaced with a *strong duality* constraint. The strong duality theorem (e.g., Bazaraa et al., 2013) states that if the problem has no duality gap, that is, some constraint qualification holds for the problem,[2] The optimal primal and dual objective values are equal. This implies that such problems can be solved to optimality by finding any solution that is both primal and dual feasible with the primal and dual objective values being equal.

Combining formulations (7) and (8), we obtain the following primal and dual constraints, combined with a strong-duality constraint:

$$C_{i3} z_i \geq a_{i3}(x) \ \ \forall i \in I \tag{9a}$$

$$C_{i3}^\top p_i - F_i z_i \leq e_{i3}(x) \ \ \forall i \in I \tag{9b}$$

$$z_i^\top F_i z_i + e_{i3}(x)^\top z_i - a_{i3}(x)^\top p_i \leq 0 \ \ \forall i \in I \tag{9c}$$

$$z_i, p_i \geq 0 \ \ \forall i \in I. \tag{9d}$$

The strong duality constraint (9c) states that the objective value of each bottom-level primal (minimisation) problem must not be higher than the value of the dual (maximisation) problem. Recall that the weak duality theorem (Bazaraa et al., 2013) states that the objective value of any solution of a minimisation problem is greater or equal to any objective value of the corresponding dual problem. This result allows us to write the strong duality constraint in an inequality form, following the approach in Huppmann and Egerer (2015), thus avoiding a quadratic equality constraint that would render a non-convex feasible region. Since the matrices F_i are PSD, constraints (9c) are convex. Knowing that weak duality guarantees the left-hand side of each constraint (9c) to be non-negative also allows us to combine the $|I|$ constraints into one by taking a sum over the left-hand side values, reducing the number of constraints.

By combining the middle-level problem (4a)–(4c) with the bottom-level problem reformulation (9), we obtain the resulting bilevel MPPDC formulation of

(4):

$$\min_{y,z_i,p_i} \quad d_2^\top y + \sum_{i\in I} e_{i2}^\top z_i \tag{10a}$$

$$\text{s.t.} \quad A_2x + B_2y + \sum_{i\in I} C_{i2}z_i \le a_2 \tag{10b}$$

$$C_{i3}z_i \ge a_{i3}(x) \ \ \forall i \in I \tag{10c}$$

$$C_{i3}^\top p_i - F_i z_i \le e_{i3}(x) \ \ \forall i \in I \tag{10d}$$

$$\sum_{i\in I}(z_i^\top F_i z_i + e_{i3}(x)^\top z_i - a_{i3}(x)^\top p_i) \le 0 \tag{10e}$$

$$y \ge 0 \tag{10f}$$

$$z_i, p_i \ge 0 \ \ \forall i \in I. \tag{10g}$$

The objective function (4a) and constraint (4b) have been modified from (4) by adding a sum over the set I to highlight the fact that we consider $|I|$ sets of decision variables z_i.

The last step is to take the (KKT) optimality conditions of the middle-level MPPDC problem (10) and add them to the top-level problem, resulting in a (trilevel) mathematical program with complementarity from primal and dual constraints. Similar to the LCP-based reformulation summarised in Section 2.3, this strong duality reformulation has the requirement that the bottom level is not directly influenced by the middle-level decision variables. With a weak trilevel structure (as per Definition 2.1), both the objective function and constraints are convex (or affine) and the KKT conditions of (10) are thus sufficient for optimality. However, to the best of our knowledge, no constraint qualification is known to hold for the problem (10). For example, Slater's constraint qualification (all non-linear constraints can be satisfied as strict inequalities, Slater, 1950) is not satisfied because weak duality states that

$$\sum_{i\in I}(z_i^\top F_i z_i + e_{i3}^\top z_i - a_{i3}^\top p_i) \ge 0$$

and thus, the non-linear strong duality constraint (10e) cannot be strictly satisfied. This means that the KKT conditions of this problem are only sufficient but not necessary for optimality. Nevertheless, this tells us that if we find a point that satisfies the KKT conditions, that point is optimal for the problem (10). The complete single-level strong duality reformulation is thus

$$min_{x,y,z} \quad c_1^\top x + d_1^\top y + e_1^\top z \tag{11a}$$

$$\text{s.t.} \quad A_1x + B_1y + C_1z \ge a_1 \tag{11b}$$

$$0 \le y \perp d_2 + B_2^\top \gamma \ge 0 \tag{11c}$$

$$0 \le \gamma \perp a_2 - A_2x - B_2y - \sum_{i\in I} C_{i2}z_i \ge 0 \tag{11d}$$

$$0 \le z_i \perp e_{i2} + C_{i2}^\top \gamma - C_{i3}^\top p_i^b - F_i^\top z_i^b + (F_i + F_i^\top) z_i \epsilon + e_{i3}(x)^\top \epsilon \ge 0 \ \ \forall i \in I \tag{11e}$$

$$0 \le p_i \perp C_{i3}z_i^b - a_{i3}(x)\epsilon \ge 0 \ \ \forall i \in I \tag{11f}$$

$$0 \le z_i^b \perp e_{i3}(x) - C_{i3}^\top p_i + F_i z_i \ge 0 \ \ \forall i \in I \tag{11g}$$

$$0 \le p_i^b \perp C_{i3}z_i - a_{i3}(x) \ge 0 \ \ \forall i \in I \tag{11h}$$

$$0 \le \epsilon \perp -\sum_{i\in I}(z_i^\top F_i z_i + e_{i3}(x)^\top z_i - a_{i3}(x)^\top p_i) \ge 0, \tag{11i}$$

where p_i^b and z_i^b are the dual variables of the bottom-level primal and dual constraints (10c) and (10d), respectively, and ϵ is the dual variable of the strong duality constraint (10e). That is, p_i^b can be interpreted as the middle-level shadow prices associated with the bottom-level primal constraints. On the other hand, it is well known that the dual of the dual problem is the primal problem, and the dual variables associated with dual constraints are the primal variables. The value of these bottom-level primal variables must be the same for the middle- and bottom-level players, i.e., $z_i^b = z_i$. Note that the right-hand sides of constraints (11e), (11f), and (11i) contain bilinear terms (assuming $e_{i3}(x)$ and $a_{i3}(x)$ are affine) including the top-level variables x, making the resulting model non-convex in general. As discussed before, such constraints can be modelled as quadratic SOS1 constraints and solved using spatial branch-and-bound-based methods.

If the problem instead has a strong trilevel structure, some of the terms a_{i3} or e_{i3} would effectively be functions of y, and the strong duality constraint (10e) would consequently have non-convex bilinear terms. The middle-level variables would be considered fixed for the bottom-level problems, but not for the middle level. A non-convex strong duality constraint in the middle-level problem (10) would result in the KKT conditions of the problem not even being sufficient for optimality. If we assume for example that the middle-level variables y appeared in linear terms added to the constant terms a and e, constraint (10e) would become

$$\sum_{i\in I}(z_i^\top F_i z_i + \left(e_{i3} + N_y^{obj}y\right)^\top z_i - \left(a_{i3}(x) + N_y^{con}y\right)^\top p_i) \le 0,$$

resulting in bilinear terms $(N_y^{obj}y)^\top z$ and $(N_y^{con}y)^\top p$, where N_y^{obj} and N_y^{con} are the coefficient matrices of the y-variables in the bottom-level objective and constraints, respectively.

2.5. Comparison of trilevel formulations

In the reformulation (6) (Gabriel, et al., 2022), the vector $\tilde{z}$ contains both the primal and dual variables of each bottom-level problem. This is because the variable $\tilde{z}$ appears in the LCP (5), which, in the problems presented in this article, represents the concatenated KKT conditions of the bottom-level problems. We denote by n_2 the number of variables

in the middle-level problem and by m_2 the number of constraints in the same problem, and analogously, n_3 and m_3 for the number of variables and constraints, respectively, in the bottom-level problem. There are then $n_2 + m_2 + 4(n_3 + m_3)$ complementarity constraints (6e)–(6g) and (6i) in formulation (6), and one variable for each complementarity constraint. Additionally, there are $n_3 + m_3 + 1$ equality constraints (6h) used in the reformulation, and the constraints (6b) for the top-level problem.

The novel strong duality formulation (11) (assuming only inequality constraints and non-negative variables in the middle- and bottom-level problems for comparison) results in $n_2 + m_2$ complementarity constraints for the middle-level variables and constraints, $2(n_3 + m_3)$ complementarity constraints for the bottom-level primal and dual variables and constraints, and one complementarity constraint for the strong duality. Because strong duality is represented as an inequality constraint, no equality constraints are needed for the strong duality reformulation.

The strong duality reformulation of the bottom level results in half the number of complementarity constraints compared to the LCP reformulation presented in Gabriel, et al. (2022), plus one for strong duality, and no equality constraints. While the LCP reformulation results in two nested optimisation problems, the intermediate MPPDC (10) in the strong duality reformulation is a single problem, explaining the difference in the number of constraints. This is computationally beneficial, as large numbers of complementarity constraints contribute greatly to the computational challenges with equilibrium problems. Additionally, it should be noted that the column-and-constraint generation algorithm (Dvorkin et al., 2018) requires the middle- and bottom-level problems to be represented as a single optimisation problem, suggesting that the strong duality approach could be easily extended to that context, unlike the LCP solution set reformulation.

On the other hand, the main disadvantage of our strong duality formulation is that the strong duality constraint (10e) retains the quadratic term from the bottom-level objective function, while the previous formulation has only affine constraints. This results in the formulation (10) not satisfying a constraint qualification, making the KKT conditions only sufficient for optimality. Additionally, unlike the strong duality formulation, the formulation in Gabriel, et al. (2022) is applicable to settings where the bottom-level complementarity conditions are not derived as KKT conditions of an optimisation problem. For example, the spatial price equilibrium problem in Gabriel, et al. (2022) could not be reformulated using strong duality.

3. Applications in energy-environmental planning

In this section, we describe a trilevel power market equilibrium model that contains environmental considerations for the top-level regional policy-maker. Finding effective instruments for emission reduction and climate change mitigation is becoming increasingly important, and we focus our attention on carbon tax (see Köppl and Schratzenstaller, 2023, for a review). At the middle level, a single regional system operator is responsible for operating transmission lines $a \in A$ between nodes $k \in K$, maximising its profit from operating the system.

At the bottom level, each energy producer $i \in I$ produces electricity at nodes $k \in K$ using energy sources $j \in J$ and sells the electricity to nodes $k' \in K$, that is, the electricity is not necessarily sold to the same node it is produced in. The producers maximise their profit from selling electricity, knowing that their decisions will affect the selling prices, making the bottom level a Cournot oligopoly. Instead of considering a fixed demand that must be satisfied exactly, we model the demand side as reacting with an affine relationship between production and price so that total demand increases linearly as the price of electricity decreases. This means, e.g., that if the producers started to generate unreasonably large amounts of electricity, the price would go down because more and more of the (elastic) demand is satisfied.

Finally, we consider a set D of representative days (Poncelet et al., 2017) of renewable generation availability factors and demand curves. The top-level regulator chooses a tax and minimum renewable share which apply for all days. In contrast, the operational decisions at the system operator and producer levels can differ between the days $d \in D$. The weights of the representative days are denoted with P_d, with $\sum_{d \in D} P_d = 1$, that is, P_d represents the fraction of days in a year that is represented by day d. The purpose of representative days is to reduce the size and complexity of the model while still being able to realistically convey the variability in renewable energy availability and demand, and they are used in models such as US-REGEN (Young, 2020) and LIMES-EU (Nahmmacher et al., 2014).

Our illustrative example is based on the model in Hobbs (2001). This model is chosen because of its simple nature, as using a more realistic model would require further discussion on assumptions and data, shifting the focus away from the methodological contributions of this article. We highlight

however, that while this reference model is a simplified representation of reality, models containing an equivalent structure to that in Hobbs (2001) are used in case studies by, e.g., Keles et al. (2020) and Ribó-Pérez et al. (2019).

3.1. The top-level regulator

On top of this trilevel hierarchy is the regional regulator which tries to both maximise the amount of electricity produced and minimise the carbon dioxide emissions from doing so. The motivation for this setting is to balance the utility from electricity generation and to maintain reasonable electricity prices, while simultaneously mitigating negative environmental outcomes.

In addition to maximising production, the regulator wants to minimise the total emissions $\sum_{ijk} \eta_{ijk} z_{ijkd}$ from electricity generation, where η_{ijk} is the emissions factor corresponding to the production level z_{ijkd}. For carbon-emitting energy sources, $\eta_{ijk} > 0$, while it is zero for zero-emission energy sources. These two objectives are then converted into a single objective by giving the total production value a weight $r \in (0, 1)$ and the total emissions a weight $(1 - r)$. By varying the value of this weight parameter, one could, for example, consider different priorities between these two objectives.

The top-level player decides on a carbon tax x, which affects each firms' variable costs: $\gamma_{ijk} = \nu_{ijk} + \eta_{ijk} x$, where $\nu_{ijk} > 0$ is the cost specific to the firm-fuel combination (i, j) in node k, and η_{ijk} is the emissions factor. Additionally, the top-level player can impose a minimum renewable share ρ that the system operator must satisfy at each node $k \in K$. We assume ρ to be the same for all nodes, but it would be straightforward to extend our model to consider this minimum renewable share to differ by node. The carbon tax and minimum renewable share affect the optimal solutions of the middle- and bottom-level players, resulting in different values for z, and consequently, the top-level objective value. Increasing the carbon tax results in lower emissions as the high-emission sources become more expensive for the producers. However, this also results in the market equilibrium in the lower levels shifting towards lower total production and higher electricity prices.

Given the upper-level variables x and ρ, the overall problem for this top-level player is given as

$$\max_{x, \rho, y, z} \quad \sum_{d \in D} P_d \sum_{i \in I, j \in J, k \in K} (r - (1 - r)\eta_{ijk}) z_{ijkd} \tag{12a}$$

$$\text{s.t.} \quad x, \rho \geq 0 \tag{12b}$$

$$z \text{ and } y \text{ solve (13) for all } d \in D. \tag{12c}$$

3.2. Profit-maximising system operator

At the middle level, following the model in Hobbs (2001), we consider a profit-maximising independent system operator (ISO). This ISO is responsible for operating the transmission lines $a \in A$ between nodes $k \in K$ for each representative day $d \in D$ and has to make sure that the lines function within their capacity limits, between $-T_a^-$ and T_a^+. The ISO chooses each node's net import y_{kd} of electricity through the transmission lines (i.e., negative y_{kd} implies that more electricity is produced than used in node k, and electricity is exported to other nodes). The line flows are determined from these using power transmission distribution factors (see, e.g., Burr Metzler (2000) for a thorough description).

The ISO's problem for the representative day $d \in D$ can be stated as the following linear program.

$$\max_{y_{kd}, z_{ijkd}} \quad \sum_{k \in K} w_{kd} y_{kd} \tag{13a}$$

$$\text{s.t.} \quad -\sum_{k \in K} PTDF_{ka} y_{kd} \leq T_a^- \quad (\phi_{ad}^-) \ \forall a \in A \tag{13b}$$

$$\sum_{k \in K} PTDF_{ka} y_{kd} \leq T_a^+ \quad (\phi_{ad}^+) \ \forall a \in A \tag{13c}$$

$$\sum_{i \in I, j \in R} z_{ijkd} \geq \rho \sum_{i \in I, j \in J} z_{ijkd} \quad (\psi_{kd}) \ \forall k \in K \tag{13d}$$

$$z_{ijkd} \text{ solve (14) for all } i \in I, \tag{13e}$$

where w_{kd} is a congestion-based wheeling fee for node $k \in K$ in day $d \in D$ and $R \subseteq J$ is the set of renewable energy sources. The wheeling fee is the unit price the producers have to pay to the ISO for selling electricity at node k, and the price that the ISO pays to the producer for each unit of electricity produced at node k, and the prices of buying and selling electricity in a node are assumed to be the same. The variables in parentheses to the right of each constraint are the corresponding dual variables.

Constraint (13d) states that the ISO has to choose such transmission values that the renewable production share in each node is at least ρ, decided by the top-level regulator. We assume that the ISO has no mechanism for influencing the producers to, for example, increase their renewable share. This assumption results in a weak trilevel structure (Definition 2.1). Instead of directly influencing the producers, the optimistic bilevel assumption described earlier results in the ISO "choosing" the best (in terms of (13a)) equilibrium solution for the bottom-level problems that satisfies (13d).

3.3. Oligopoly of the producers

We next consider the lower-level optimisation problems for a set of energy firms $i \in I = \{1, ..., n_F\}$. We start by presenting these problems formulated for a bilateral market where electricity producers sell directly to consumers, which turns out to be the simpler case, and then proceed to add arbitragers to arrive at a POOLCO market model where the producers instead sell their electricity to a central auction. The POOLCO model more accurately represents the Nordic system and is thus used in the case study in Section 4.2. For a detailed discussion on different market types, we refer the reader to Ilic et al. (1998).

Let us first assume that at this lower level, these n_F firms constitute the entire market. Each firm has a production capacity in some of the nodes $k \in K$ and can bilaterally sell their electricity directly to any of the nodes. For production, the producers have a set of energy sources $j \in J$. Our formulation for this producer level follows the ideas in Hobbs (2001).

In this first model without arbitragers, every firm $i \in I$ decides on its sales and production for each node $k \in K$ and day $d \in D$, taking into account linear inverse demand functions $p_{kd}(s_{1kd}, ..., s_{n_F kd}) = \alpha_{kd} - \beta_{kd} \sum_{i=1}^{N_F} s_{ikd}$ with price intercept $\alpha_{kd} > 0$ and slope $\beta_{kd} > 0$. These parameters are assumed to vary per day, representing the changes in demand. Recall that s_{ikd} is the amount of electricity sold by producer i to node k in day d, and the market price at node k thus depends on the sum of the sales of all firms into node k.

Additionally, each producer $i \in I$ has maximum production levels $z_{ijkd}^{\max}$ determined by their installed production capacity. For wind and solar power, the maximum production level depends on the representative day d. Each producing firm solves the profit-maximisation problem

$$\max_{s_{ikd},\, z_{ijkd}} \quad \sum_{k \in K} \left(\left(\alpha_{kd} - \beta_{kd} \sum_{i' \in I} s_{i'kd} \right) s_{ikd} - \sum_{j \in J} \gamma_{ijk} z_{ijkd} - (s_{ikd} - z_{ijkd}) w_{kd} \right) \tag{14a}$$

$$\text{s.t.} \quad z_{ijkd} \le z_{ijkd}^{\max} \quad (\lambda_{ijkd}) \ \ \forall j \in J, k \in K \tag{14b}$$

$$\sum_{k \in K} s_{ikd} = \sum_{j \in J,\, k \in K} z_{ijkd} \quad (\theta_{id}) \tag{14c}$$

$$z_{ijkd}, s_{ikd} \ge 0, \tag{14d}$$

where γ_{ijk} is the marginal production cost for firm i in node k with fuel type j, composed as the sum of a firm-specific cost ν_{ijk} and an emissions cost $\eta_{ijk} x$, depending on the carbon tax x determined by the regulator.

The first term in (14a), involving the sales variables s_{ikd} represents the revenue from selling electricity to different nodes $k \in K$. The nodal price is $p_{kd} = \alpha_{kd} - \beta_{kd} \sum_{i \in I} s_{ikd}$. The cost of producing energy is γ_{ijk}. The producers pay a wheeling fee w_{kd}, which is determined by the transmission network congestion and paid to the ISO. In this hub-network model, the wheeling fee is also what the ISO pays the producers for producing extra energy in each node k.

Constraint (14b) states that production cannot exceed capacity $z_{ijkd}^{\max}$ and constraint (14c) states that for each producer, the total sales must equal total production. It is easy to see that the objective function (14a) is concave for $\beta_{kd} > 0$ and the constraints are affine. Thus, the bottom-level problem (14) has the same structure as the quadratic problems discussed in Section 2.4.

Finally, we include a market-clearing constraint

$$\sum_{i \in I} s_{ikd} - \sum_{i \in I,\, j \in J} z_{ijkd} = y_{kd} \quad (w_{kd}) \ \ \forall k \in K, d \in D. \tag{15}$$

This constraint is similar to constraint (14c), which instead considers the difference between sales and production for each producer $i \in I$. We adopt the Bertrand assumption used in Hobbs (2001): the system operator sees the wheeling fees as fixed, instead of using market power to affect their values. In order to achieve this, the market-clearing constraint (15) is considered outside the system operator and producer problems, appearing "separately" in the final single-level formulation, effectively becoming a top-level constraint.

3.3.1. Extending the producer oligopoly: including arbitrage

We are interested in modelling the Nordic market and, to achieve that, we extend the bilateral market model represented by (14) into a POOLCO model. In a POOLCO market model, it is assumed that the producers sell their electricity to a central auction where the price is determined based on the amount of sold electricity and network congestion. Burr Metzler (2000) and Hobbs (2001) show that a bilateral market with *arbitragers* is equivalent to a POOLCO market, assuming Cournot competition. Arbitragers are bottom-level players who have no production capacity, but they instead make their profits by exploiting the price differences between nodes, buying cheap electricity and selling it to nodes with a higher price. They act as price-takers and thus do not anticipate their effect on the price p_{kd}. The arbitrager's problem is

$$\max_{a} \quad \sum_{k \in K} (p_{kd} - w_{kd}) a_{kd} \tag{16a}$$

$$\text{s.t.} \quad \sum_{k\in K} a_{kd} = 0 \quad (p_d^H), \tag{16b}$$

where a_{kd} is the amount of electricity sold by the arbitrager to node k in day d and the price at node $k \in K$ depends on the sales from the producers and the arbitragers, thus becoming $p_{kd} = \alpha_{kd} - \beta_{kd}(\sum_{i\in I, j\in J} s_{ikd} + a_{kd})$. We can trivially obtain the KKT conditions of (16), a linear maximisation problem (recall that the arbitragers are price-takers, and p_{kd} is thus treated as a constant). The KKT conditions (17d) and (17e) are necessary and sufficient for optimality and adding them to (14), we obtain

$$\max_{s_{ikd}, z_{ijkd}, a_{ikd}} \quad \sum_{k\in K}\left(\left(\alpha_{kd} - \beta_{kd}\left(\sum_{i'\in I} s_{i'kd} + a_{ikd}\right)\right)s_{ikd} - \sum_{j\in J}\gamma_{ijk}z_{ijkd} - (s_{ikd} - z_{ijkd})w_{kd}\right) \tag{17a}$$

$$\text{s.t.} \quad z_{ijkd} \le z_{ijkd}^{\max} \quad (\lambda_{ijkd}) \ \forall j \in J, k \in K \tag{17b}$$

$$\sum_{k\in K} s_{ikd} = \sum_{j\in J,\, k\in K} z_{ijkd} \quad (\theta_{id}) \tag{17c}$$

$$\alpha_{kd} - \beta_{kd}\left(\sum_{i'\in I} s_{i'kd} + a_{ikd}\right) = p_{id}^H + w_{kd} \ \forall k \in K \tag{17d}$$

$$\sum_{k\in K} a_{ikd} = 0 \tag{17e}$$

$$z_{ijkd}, s_{ikd} \ge 0, \tag{17f}$$

where a_{ikd} is the net amount of power sold in node k by the arbitrager(s), and p_{id}^H, the dual variable associated with the arbitrager constraint, is the price at the central auction H. Both a_{ikd} and p_{id}^H are indexed over the different producers $i \in I$, to highlight that each producer can influence these values with their decisions and to avoid decision variables shared by players. This would result in a generalised Nash equilibrium problem (Facchinei and Kanzow, 2010) that would be computationally more challenging. However, the values a_{ikd} and p_{id}^H are the same for all producers at equilibrium, as shown in Appendix C, and the approach of having separate variables for each producer is thus valid. Constraint (17d) can be therefore written as $p_{kd} - w_{kd} = p_{id}^H$. That is, including arbitragers results in the producers selling their electricity to the central auction at the hub price p_{id}^H (or simply p_d^H at equilibrium), which is the sum of the price p_{kd} at node $k \in K$ and the wheeling fee w_{kd} paid to the system operator. Constraint (17e) states that since the arbitragers have no production capacity, their net sales amounts must be zero. The objective function is still concave after adding the arbitrage variables, and the new constraints are affine. Burr Metzler (2000) shows further substitutions and simplifications to the producer and system operator problems, which are shown in Appendix C, along with the resulting model that is used for the computational experiments in Section 4.

4. Computational experiments

To illustrate the performance of the trilevel optimisation framework in a realistic problem setting, we solve the trilevel model described in the previous section, using randomly generated instances of varying sizes. The data used in these computational experiments mimic the data in the case study of Belyak et al. (2023), whose data are from the ENTSO-E Transparency Platform (Hirth et al., 2018) and are further described in Section 4.2. The computational experiments were performed using eight CPU threads and 16GB of RAM. All code were implemented in Julia v1.7.3 (Bezanson et al., 2017) using the Gurobi solver v10.0.0 (Gurobi Optimization, LLC, 2022) and JuMP v1.5.0 (Dunning et al., 2017) and are available in (github.com/gamma-opt/trilevel-energy).

4.1. Comparing formulations

We compare the performance of the two single-level reformulations, the LCP-based reformulation from Gabriel, et al. (2022) (Section 2.3) and our strong duality reformulation (Section 2.4) by solving 50 randomly generated problems with two producers, five energy sources, three nodes, and three representative days. This problem size was chosen as the base case because it seems to be large enough to make the problems challenging to solve, but small enough for them to be mostly solvable within a time limit of 1 h.

The results are presented in Figure 1, and the main observation here is that the novel strong duality formulation is faster in most cases. In Figure 1, markers below the diagonal (dashed line) correspond to such cases. In 13 instances, the formulation of Gabriel, et al. (2022) did not find an optimal solution in an hour while our strong duality formulation did. One major issue with both models compared here is that usually the first feasible solutions are found at the end of the solution process, and most of the solution time is spent on improving the dual bound without finding any feasible solutions. Nevertheless, changing solver parameters to emphasise finding feasible solutions was not found to have a major impact on performance.

As discussed in Section 2.5, the strong duality formulation results in fewer constraints than the reformulation in Gabriel, et al. (2022). Recall that in our models, complementarity constraints are

formulated as SOS1 constraints. The model sizes in the base case test problems are presented in Table 1, and out of the two, our strong duality model is smaller, except for having one more quadratic SOS1 constraint to represent the strong duality constraint (11i). In our model, all non-complementarity quadratic constraints are inequality constraints, while in the LCP-based model, there is one quadratic equality constraint (the first one in (6h)).

Next, we analyse how problem size affects solution times by varying either the number of producers, energy sources, nodes, and representative days from the base case, one parameter at a time. The results are presented in Figure 2. The medium-sized cases in each subfigure are similar to each other, which is expected as the problem sizes are the same. Varying the number of producers or energy sources seems to have only a small effect on the solution times while changing the number of nodes has a far stronger effect. The effect of the number of representative days is stronger than that of the number of producers and energy sources but seems to be weaker than that of the number of nodes.

We can also see that the number of problems that were not solved to optimality within the time limit is affected by the number of nodes and representative days, but not by the number of firms or energy sources. Additionally, the novel strong duality model finds an optimal solution more frequently than the previous formulation. As predicted in Section 2.4, the larger number of complementarity constraints in the LCP formulation (Table 1) proves to be computationally challenging, and the smaller strong duality model is solved faster.

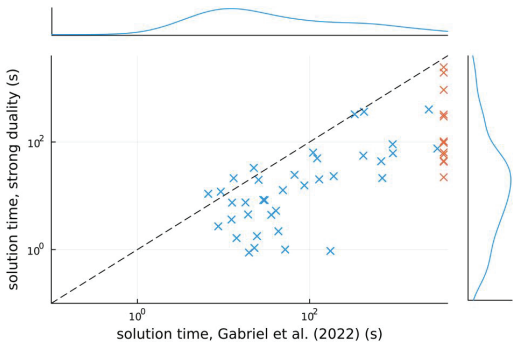


Figure 1. Solution times for the two formulations on 50 random instances with 2 producers, 5 energy sources, 3 nodes, and 3 representative days. If one of the methods failed to find a solution within 3600 s, an orange marker is used, and the marginal distributions on the right and top sides exclude unsolved instances.

4.2. Case study: a five-node Nordic energy system

The case study in Belyak et al. (2023) considers five nodes, representing Finland, Sweden, Norway, Denmark, and the combined Baltic countries (Estonia, Latvia, and Lithuania). There are five producers, each owning production capacity in one of the five nodes. Nine different energy sources are available, consisting of five conventional sources: nuclear, coal, gas (closed- and open-cycle) and biomass, and four renewable sources: solar, hydro, onshore, and offshore wind. Additionally, we consider three representative days of renewable generation availability factors and demand curves. Recall that in our model, the top-level regulator makes their decisions independent of the day considered, that is, the carbon tax and minimum renewable share are constant across different representative days. These representative days are obtained in Belyak et al. (2023) by performing hierarchical clustering on demand, price, and renewable availability data.

Day 1 is a winter day with higher demand, low solar availability, and medium wind availability. Days 2 and 3 have a lower demand with day 2 representing a windy day with medium solar availability, and day 3 representing a sunny day with low wind availability. The details of the hierarchical clustering process can be found in Belyak et al. (2023).

In Figure 3, the production portfolio (a weighted average over the representative days) is presented for a model with no carbon tax (i.e., the regulator heavily prefers maximising production over minimising emissions) and a carbon tax of 23 €/ton (enough to remove nearly all emissions). Compared to the baseline with no carbon tax, this 23 €/ton tax decreases the total production by 2.8%. In this example, these carbon tax values are achieved by setting the weight parameter r in the top-level objective (12a) to 0.8 and 0.4, respectively.

Because of the substantial hydropower production capacity in the Nordic system, particularly in Norway and Sweden (IRENA, 2023), the renewable

Table 1. Model sizes for the two reformulations.

	Strong duality (this article)	LCP (Gabriel et al., 2022)
Variables	678	949
Affine constraints	306	757
Quadratic constraints	100	100
Affine SOS1	288	648
Quadratic SOS1	100	99

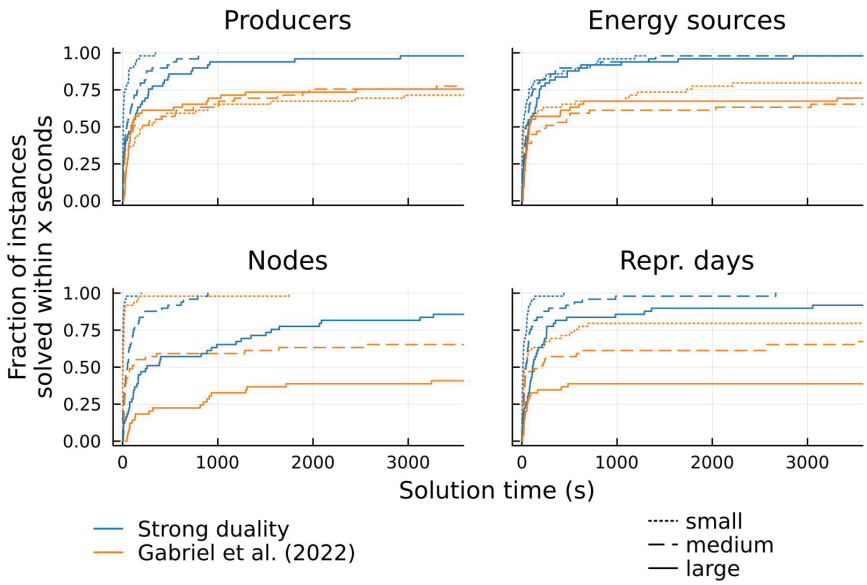
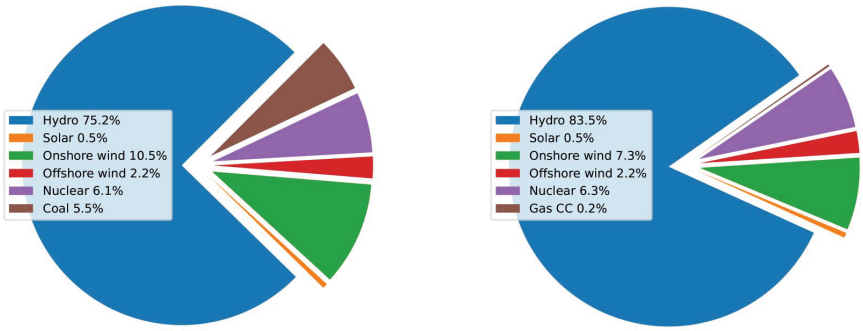


Figure 2. Cumulative distribution functions of solution times for the two formulations with 1–3 producers, 4–6 energy sources, and 2–4 nodes and representative days. For each problem size, 50 instances are generated and solved.



(a) No carbon tax. (b) Carbon tax 23e/ton.

Figure 3. Weighted average electricity production portfolio over the five nodes and three representative days.

share of production is large even without a carbon tax. A part of the increase in hydropower usage when the carbon tax is introduced comes from decreasing onshore wind production. This is a consequence of the simplified nature of the model and data, as the operational costs for both hydropower and onshore wind power are zero, resulting in multiple optima and indifference for the producers to use one or the other, as long as the production capacity of neither is exceeded. This artefact of the model could be easily removed by, e.g., setting the operational cost of either energy source to a small positive number instead of zero, causing the producers to prefer the cheaper source. However, this would imply an artificial preference for one source over the other. The only significant source of emissions is coal, and introducing a carbon tax of 23€/ton removes all coal from the portfolio, bringing in a small amount of closed-cycle gas power instead. The closed-cycle gas production occurs in the Baltics for day 1, and to understand this emergence of gas better, we must examine the transmission network in Figure 4.

The first representative day has the highest network usage with large amounts of electricity transmitted from Norway to Finland through Sweden. With the carbon tax, the importance of transmission is further highlighted as the hydropower capacity in Norway is used for lowering overall prices under high demand and low production from both solar power and high-emission sources. The differences between representative days 2 and 3 are more subtle, but we can see, e.g., the reliance on wind power in Denmark: in the low-wind day 3, the carbon tax

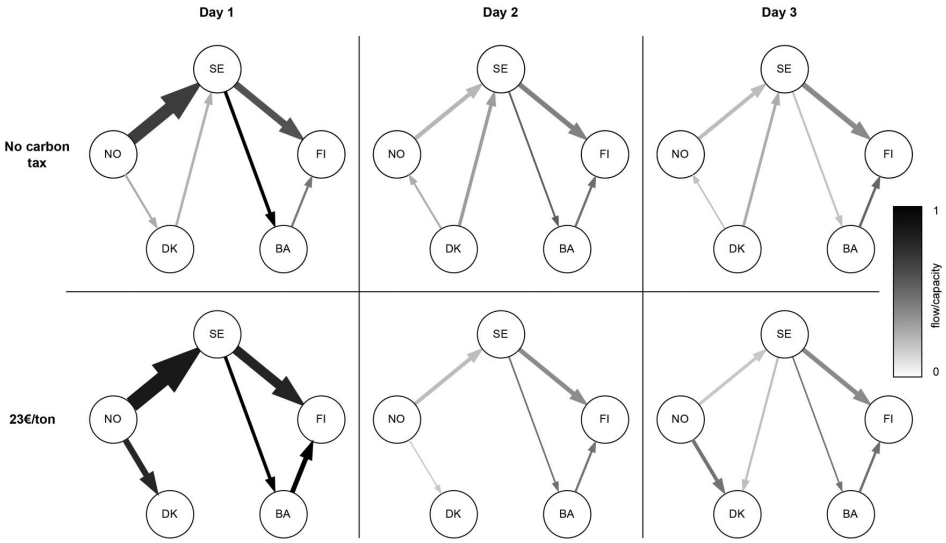


Figure 4. Transmission grid usage with different carbon taxes and representative days. The size of an arrow is proportional to the flow on the line and the colour of an arrow represents congestion: black arrows correspond to lines operating at their limit. The nodes are FI = Finland, SE = Sweden, NO = Norway, DK = Denmark, BA = Baltic countries.

results in Denmark importing a significant amount of electricity from Norway, compared to the high-wind day 2. On the first day with a carbon tax of 23 €/ton, both lines connecting the Baltic countries to the rest of the system are at their capacity, explaining why the Baltic countries start using gas power after a carbon tax is introduced. This illustrates the complex interplay between the three levels that is captured by our model.

5. Conclusions

In this article, we propose a novel formulation for trilevel optimisation problems focusing on energy systems planning with environmental considerations. Additionally, we characterise the notion of weak and strong trilevel structures and compare the computational performance of the novel strong duality-based reformulation in this article and the LCP-based reformulation in Gabriel, et al. (2022).

The computational results are encouraging, as we are able to solve the case study to optimality within a few minutes despite the fact that both single-level reformulations considered are non-convex problems. However, we note that preliminary experiments with seemingly small extensions to the model, such as adding ramping constraints (limiting the change in production between consecutive periods) to the producer problem made the problem computationally intractable. The small size of the case study is indicative of the very challenging (non-convex) nature of these problems, and the authors note that the reformulations and solution methods in this article should be viewed as one of the first steps towards an efficient solution framework for trilevel problems.

For the results in this article, an off-the-shelf solver is used, which is useful to ensure a low barrier-to-entry for using the developed formulation. However, we believe that the computational performance can be increased considerably using specialised solution methods like column-and-constraint generation (Dvorkin et al., 2018). Notably, ideas such as bilevel branch-and-bound (Fischetti et al., 2018) and convex hull reformulations of the middle-level feasible region (Santana and Dey, 2020) may be explored in the context of the problems presented in this article. In addition, the model could also be extended to consider transmission and/or production capacity expansion over multiple time periods, especially if computationally more efficient reformulations and solution methods are developed.

Despite the outstanding computational challenges, we show that the novel reformulation improves computational performance compared to the previous formulation (Gabriel, et al., 2022), and we show that the framework can be applied to a setting representing the Nordic electricity market, and results on the effect of carbon tax can be obtained. A limitation of the formulation approach presented in this article and that originally proposed by Gabriel, et al. (2022) is that they require a weak trilevel structure. In practice, relevant problems may instead have a strong trilevel structure, precluding the use of these reformulations. Thus, further research is needed on

developing (heuristic) solution methods for problems with a strong trilevel structure.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Notes

1. For completeness, the steps for obtaining the dual (8) from the primal problem (7) are presented in Appendix B.
2. For problems with only affine constraints, such as (7), the Abadie constraint qualification is always satisfied (Bazaraa et al., 2013).

Funding

The calculations presented in this paper were performed using computer resources within the Aalto University School of Science "Science-IT" project. Fabricio Oliveira and Olli Herrala were supported by the Research Council of Finland (decision numbers 348094 and 332180). Steven A. Gabriel was supported by the National Science Foundation (NSF) Award #2113891, Civil Infrastructure Systems. Tommi Ekholm was supported by the Research Council of Finland (decision number 341311).

References

Bard, J. F. (1983). An algorithm for solving the general bilevel programming problem. *Mathematics of Operations Research*, *8*(2), 260–272. https://doi.org/10.1287/moor.8.2.260

Bard, J. F. (1991). Some properties of the bilevel programming problem. *Journal of Optimization Theory and Applications*, *68*(2), 371–378. https://doi.org/10.1007/BF00941574

Bard, J. F., & Moore, J. T. (1990). A branch and bound algorithm for the bilevel programming problem. *SIAM Journal on Scientific and Statistical Computing*, *11*(2), 281–292. https://doi.org/10.1137/0911017

Baringo, L., & Conejo, A. J. (2012). Transmission and wind power investment. *IEEE Transactions on Power Systems*, *27*(2), 885–893. https://doi.org/10.1109/TPWRS.2011.2170441

Bazaraa, M. S., Sherali, H. D., & Shetty, C. M. (2013). *Nonlinear programming: Theory and algorithms*. John Wiley & Sons.

Beale, E. M. L., & Tomlin, J. A. (1970). Special facilities in a general mathematical programming system for non-convex problems using ordered sets of variables. *Operational Research*, *69*(99), 447–454.

Belyak, N., Gabriel, S. A., Khabarov, N., & Oliveira, F. (2023). Optimal transmission expansion planning in the context of renewable energy integration policies. https://doi.org/10.1016/j.jclepro.2024.141955

Bezanson, J., Edelman, A., Karpinski, S., & Shah, V. B. (2017). Julia: A fresh approach to numerical computing. *SIAM Review*, *59*(1), 65–98. https://doi.org/10.1137/141000671

Burr Metzler, C. (2000). *Complementarity models of competitive oligopolistic electric power generation markets*. [Doctoral dissertation, The Johns Hopkins University]. https://cats.informa.com/PTS/login.do https://www.proquest.com/openview/23325a82a09fd6418e0f13-c0164e2dd6/1

Camacho-Vallejo, J.-F., Corpus, C., & Villegas, J. G. (2023). Metaheuristics for bilevel optimization: A comprehensive review. *Computers & Operations Research*, *161*, 106410. https://doi.org/10.1016/j.cor.2023.106410

Cottle, R. W., Pang, J.-S., & Stone, R. E. (2009). *The linear complementarity problem*. Society for Industrial and Applied Mathematics.

Dempe, S., & Zemkoho, A. (Editors). (2020). Bilevel optimization. In: Panos M. Pardalos, My T. Thai (Series Editors), University of Florida. *Springer optimization and its applications* (Vol.161). Springer. Pages 3–26.

Dorn, W. S. (1960). Duality in quadratic programming. *Quarterly of Applied Mathematics*, *18*(2), 155–162. https://doi.org/10.1090/qam/112751

Dunning, I., Huchette, J., & Lubin, M. (2017). JuMP: A modeling language for mathematical optimization. *SIAM Review*, *59*(2), 295–320. https://doi.org/10.1137/15M1020575

Dvorkin, Y., Fernandez-Blanco, R., Wang, Y., Xu, B., Kirschen, D. S., Pandzic, H., Watson, J.-P., & Silva-Monroy, C. A. (2018). Co-planning of investments in transmission and merchant energy storage. *IEEE Transactions on Power Systems*, *33*(1), 245–256. https://doi.org/10.1109/TPWRS.2017.2705187

Facchinei, F., & Kanzow, C. (2010). Generalized Nash equilibrium problems. *Annals of Operations Research*, *175*(1), 177–211. https://doi.org/10.1007/s10479-009-0653-x

Fischetti, M., Ljubić, I., Monaci, M., & Sinnl, M. (2018). On the use of intersection cuts for bilevel optimization. *Mathematical Programming*, *172*(1–2), 77–103. https://doi.org/10.1007/s10107-017-1189-5

Gabriel, S. A., Conejo, A. J., Fuller, J. D., Hobbs, B. F., & Ruiz, C. (2012). *Complementarity modeling in energy markets*. Springer Science & Business Media.

Gabriel, S. A., Leal, M., & Schmidt, M. (2022). On linear bilevel optimization problems with complementarity-constrained lower levels. *Journal of the Operational Research Society*, *73*(12), 2706–2716. https://doi.org/10.1080/01605682.2021.2015254

Gao, S., & Liu, N. (2021). Improving the resilience of port–hinterland container logistics transportation systems: A bi-level programming approach. *Sustainability*, *14*(1), 180. https://doi.org/10.3390/su14010180

Gu, Y., Cai, X., Han, D., & Wang, D. Z. (2019). A trilevel optimization model for a private road competition problem with traffic equilibrium constraints. *European Journal of Operational Research*, *273*(1), 190–197. https://doi.org/10.1016/j.ejor.2018.07.041

Gurobi Optimization, LLC. (2022). Gurobi Optimizer Reference Manual. https://www.gurobi.com

Hájek, M., Zimmermannová, J., Helman, K., & Rozenský, L. (2019). Analysis of carbon tax efficiency in energy industries of selected EU countries. *Energy Policy*, *134*, 110955. https://doi.org/10.1016/j.enpol.2019.110955

Hirth, L., Mühlenpfordt, J., & Bulkeley, M. (2018). The ENTSO-E transparency platform–A review of Europe's most ambitious electricity data platform. *Applied Energy*, *225*, 1054–1067. https://doi.org/10.1016/j.apenergy.2018.04.048

Hobbs, B. F. (2001). Linear complementarity models of Nash-Cournot competition in bilateral and POOLCO

power markets. *IEEE Transactions on Power Systems*, *16*(2), 194–202. https://doi.org/10.1109/59.918286
Huppmann, D., & Egerer, J. (2015). National-strategic investment in European power transmission capacity. *European Journal of Operational Research*, *247*(1), 191–203. https://doi.org/10.1016/j.ejor.2015.05.056
Ilic, M., Galiana, F., & Fink, L. (1998). *Power systems restructuring: Engineering and economics*. Kluwer Academic Publishers.
IRENA (2023). *Renewable capacity statistics 2023*. International Renewable Energy Agency.
Jin, S., & Ryan, S. M. (2014). A tri-level model of centralized transmission and decentralized generation expansion planning for an electricity market—Part I. *IEEE Transactions on Power Systems*, *29*(1), 132–141. https://doi.org/10.1109/TPWRS.2013.2280085
Karush, W. (1939). *Minima of functions of several variables with inequalities as side conditions* [Master's thesis, University of Chicago].
Keles, D., Dehler-Holland, J., Densing, M., Panos, E., & Hack, F. (2020). Cross-border effects in interconnected electricity markets-an analysis of the Swiss electricity prices. *Energy Economics*, *90*, 104802. https://doi.org/10.1016/j.eneco.2020.104802
Kieffer, E., Danoy, G., Brust, M. R., Bouvry, P., & Nagih, A. (2020). Tackling large-scale and combinatorial bi-level problems with a genetic programming hyper-heuristic. *IEEE Transactions on Evolutionary Computation*, *24*(1), 44–56. https://doi.org/10.1109/TEVC.2019.2906581
Kleinert, T., Labbé, M., Ljubić, I., & Schmidt, M. (2021). A survey on mixed-integer programming techniques in bilevel optimization. *EURO Journal on Computational Optimization*, *9*, 100007. https://doi.org/10.1016/j.ejco.2021.100007
Köppl, A., & Schratzenstaller, M. (2023). Carbon taxation: A review of the empirical literature. *Journal of Economic Surveys*, *37*(4), 1353–1388. https://doi.org/10.1111/joes.12531
Kuhn, H. W., & Tucker, A. W. (1951). Nonlinear programming. *Berkeley Symposium on Mathematical Statistics and Probability*, *1951*, 481–492.
Leyffer, S., Menickelly, M., Munson, T., Vanaret, C., & Wild, S. M. (2020). A survey of nonlinear robust optimization. *Information Systems and Operational Research*, *58*(2), 342–373. https://doi.org/10.1080/03155986.2020.1730676
Nahmmacher, P., Schmid, E., Knopf, B. (2014). *Documentation of LIMES-EU - A long-term electricity system model for Europe*. Retrieved February 22, 2024, from https://www.pik-potsdam.de/en/institute/departments/transformation-pathways/models/limes/DocumentationLIMESEU_2014.pdf
Pineda, S., Bylling, H., & Morales, J. (2018). Efficiently solving linear bilevel programming problems using off-the-shelf optimization software. *Optimization and Engineering*, *19*(1), 187–211. https://doi.org/10.1007/s11081-017-9369-y
Poncelet, K., Hoschle, H., Delarue, E., Virag, A., & Drhaeseleer, W. (2017). Selecting representative days for capturing the implications of integrating intermittent renewables in generation expansion planning problems. *IEEE Transactions on Power Systems*, *32*(3), 1936–1948. https://doi.org/10.1109/TPWRS.2016.2596803
Ribó-Pérez, D., Van der Weijde, A. H., & Álvarez-Bel, C. (2019). Effects of self-generation in imperfectly competitive electricity markets: The case of Spain. *Energy Policy*, *133*, 110920. https://doi.org/10.1016/j.enpol.2019.110920
Ruiz, C., Conejo, A. J., & Smeers, Y. (2012). Equilibria in an oligopolistic electricity pool with stepwise offer curves. *IEEE Transactions on Power Systems*, *27*(2), 752–761. https://doi.org/10.1109/TPWRS.2011.2170439
Santana, A., & Dey, S. S. (2020). The convex hull of a quadratic constraint over a polytope. *SIAM Journal on Optimization*, *30*(4), 2983–2997. https://doi.org/10.1137/19M1277333
Sauma, E. E., & Oren, S. S. (2007). Economic criteria for planning transmission investment in restructured electricity markets. *IEEE Transactions on Power Systems*, *22*(4), 1394–1405. https://doi.org/10.1109/TPWRS.2007.907149
Siddiqui, S., & Gabriel, S. A. (2013). An SOS1-based approach for solving MPECs with a natural gas market application. *Networks and Spatial Economics*, *13*(2), 205–227. https://doi.org/10.1007/s11067-012-9178-y
Sinha, A., Malo, P., & Deb, K. (2018). A review on bilevel optimization: From classical to evolutionary approaches and applications. *IEEE Transactions on Evolutionary Computation*, *22*(2), 276–295. https://doi.org/10.1109/TEVC.2017.2712906
Slater, M. (1950). Lagrange multipliers revisited. *Cowles Foundation Discussion Papers*, *304*.
Still, G. (2002). Linear bilevel problems: Genericity results and an efficient method for computing local minima. *Mathematical Methods of Operations Research*, *55*(3), 383–400. https://doi.org/10.1007/s001860200189
Ye, J. J., & Zhu, D. (1995). Optimality conditions for bilevel programming problems. *Optimization*, *33*(1), 9–27. https://doi.org/10.1080/02331939508844060
Young, D. (2020). US-REGEN model documentation. Retrieved February 22, 2024, from https://www.epri.com/research/products/3002016601

Appendix A. Reformulation of a bottom-level LCP with a positive semi-definite M

Cottle et al. (2009) show that if the matrix M is positive semi-definite (PSD), all solutions to the LCP

$$0 \leq z \perp q + N_x x + N_y y + Mz \geq 0, \tag{A1}$$

can be obtained as the following polyhedral set:

$$\begin{aligned} \{z \in R^{n_z}_{\geq 0} :\ & q + N_x x + N_y y + Mz \geq 0, \\ & (q + N_x x + N_y y)^T (z - \bar{z}) = 0, \\ & (M + M^T)(z - \bar{z}) = 0\}, \end{aligned} \tag{A2}$$

where $\bar{z}$ is a solution to the LCP.

Hence, the middle-level problem can be rewritten as

$$min_{y,z\geq 0} \quad d_2^\top y + e_2^\top z \tag{A3a}$$

$$s.t. \quad A_2 x + B_2 y + C_2 z \geq a_2 \tag{A3b}$$

$$q + N_x x + N_y y + Mz \geq 0 \tag{A3c}$$

$$(q + N_x x + N_y y)^\top (z - \bar{z}) = 0 \tag{A3d}$$

$$(M + M^\top)(z - \bar{z}) = 0. \tag{A3e}$$

We observe that (A3d) includes a bilinear term $y^\top N_y^\top z$ in an equality constraint. This is a non-convex constraint, precluding the direct use of KKT conditions for obtaining an optimal solution to (A3). However, for problems with a weak trilevel structure, $N_y = 0$ and these bilinear terms vanish. In the next theorem, we assume $N_y = 0$.

Theorem A.1. *Let M be a positive semi-definite matrix. Then, (x^*, y^*, z^*) is an optimal solution of Problem (3) with middle level (4) if and only if $(x^*, y^*, z^*, \bar{z}^*)$ is an optimal solution of the problem*

$$\min_{x,y,z,\bar{z}} \quad c_1^\top x + d_1^\top y + e_1^\top z \tag{A4a}$$

$$s.t. \quad A_1 x + B_1 y + C_1 z \geq a_1 \tag{A4b}$$

$$\begin{aligned} \bar{z} \in \arg\min_{z' \geq 0} \{ & z'^\top (q + N_x x + M z') : \\ & q + N_x x + M z' \geq 0 \quad (\beta) \}, \end{aligned} \tag{A4c}$$

$$\begin{aligned} y, z \in \arg\min_{\hat{y}, \hat{z} \geq 0} \{ & d_2^\top \hat{y} + e_2^\top \hat{z} : \\ & q + N_x x + M \hat{z} \geq 0 \quad (\delta) \\ & (q + N_x x)^\top (\hat{z} - \bar{z}) = 0 \quad (\zeta) \\ & (M + M^\top)(\hat{z} - \bar{z}) = 0 \quad (\eta) \\ & A_2 x + B_2 \hat{y} + C_2 \hat{z} \geq a_2 \quad (\gamma) \}, \\ \text{such that } & (\bar{z}^*)^\top (q + N_x x^* + M \bar{z}^*) = 0. \end{aligned} \tag{A4d}$$

See Theorem 6 in Gabriel et al. (2022) for a proof of this result as well as related theoretical aspects of the general form of the problem.

The two nested optimisation problems in (A4) are a convex QP (A4c) and an LP (A4d). Hence, the KKT conditions of both problems are necessary and sufficient for optimality and the two inner problems can be replaced by their necessary and sufficient KKT conditions, leading to the single-level reformulation (6).

Appendix B. Formulating the dual of a QP with affine constraints

Given a quadratic program with affine constraints

$$\min_{z_i} \quad \frac{1}{2} z_i^\top F_i z_i + e_{i3}(x)^\top z_i \tag{B1a}$$

$$\text{s.t.} \quad C_{i3} z_i \geq a_{i3}(x) \quad (p_i) \tag{B1b}$$

$$z_i \geq 0 \quad (s_i), \tag{B1c}$$

where we assume F_i is a positive semi-definite (PSD) symmetric matrix, the Lagrangian of the problem is

$$\begin{aligned} L(z_i, s_i, p_i) = & \frac{1}{2} z_i^\top F_i z_i + e_{i3}(x)^\top z_i + p_i^\top (a_{i3}(x) - C_{i3} z_i) \\ & - s_i^\top z_i, \end{aligned} \tag{B2}$$

where p_i and s_i are non-negative Lagrange multipliers or dual variables. The first-order optimality condition is thus

$$\nabla_{z_i} L(z_i, s_i, p_i) = F_i z_i + e_{i3}(x) - C_{i3}^\top p_i - s_i = 0 \tag{B3}$$

and rearranging (B2) gives us

$$\frac{1}{2} z_i^\top F_i z_i - z_i^\top C_{i3}^\top p_i + z_i^\top e_{i3}(x) - z_i^\top s_i + a_{i3}(x)^\top p_i, \tag{B4}$$

which, using the first-order condition $-F_i z_i = -C_{i3}^\top p_i + e_{i3}(x) - s_i$, becomes

$$-\frac{1}{2} z_i^\top F_i z_i + a_{i3}(x)^\top p_i. \tag{B5}$$

Maximising Eq. (B5), subject to the first-order optimality condition for z_i and treating s_i as a slack variable and removing its explicit representation from the problem results in the Lagrangian dual formulation

$$\max_{p_i, z_i} \quad -\frac{1}{2} z_i^\top F_i z_i + a_{i3}(x)^\top p_i \tag{B6a}$$

$$\text{s.t.} \quad C_{i3}^\top p_i - F_i z_i \leq e_{i3}(x) \tag{B6b}$$

$$p_i \geq 0. \tag{B6c}$$

Appendix C. Further substitutions for the middle and bottom levels

The producer model can be further simplified using the substitution $s_{ikd} = \sum_{j \in J} z_{ijkd}$, removing the sales variables and the balance constraint (17c). For a further reduction, the remaining equality constraints (17d) and (17e) can be used to solve for a and p^H.

We have the necessary and sufficient KKT conditions

$$\alpha_{kd} - \beta_{kd} \left(\sum_{i' \in I} s_{i'kd} + a_{ikd} \right) = p_{id}^H + w_{kd} \forall k \in K \tag{C1}$$

$$\sum_{k \in K} a_{ikd} = 0 \tag{C2}$$

of the arbitrager's problem, and with the substitution $\sum_{j \in J} z_{ijkd} = s_{ikd}$, we get

$$\alpha_{kd} - \beta_{kd} (Z_{kd} + a_{ikd}) = p_{id}^H + w_{kd} \forall k \in K \tag{C4}$$

$$\sum_{k \in K} a_{ikd} = 0, \tag{C5}$$

where $Z_{kd} = \sum_{i \in F, j \in J} z_{ijkd}$. In matrix form, we get

$$\begin{bmatrix} Q_d & \mathbf{1} \\ \mathbf{1}^\top & 0 \end{bmatrix} \begin{bmatrix} a_{id} \\ p_{id}^H \end{bmatrix} = \begin{bmatrix} \alpha_d - Q Z_d - w_d \\ 0 \end{bmatrix}, \tag{C7}$$

where Q_d is a square diagonal matrix with the element on the kth row and column being β_{kd} and $\mathbf{1}$ is a vector of ones. It can be shown that

$$\begin{bmatrix} Q_d & \mathbf{1} \\ \mathbf{1}^\top & 0 \end{bmatrix}^{-1} = \begin{bmatrix} L^d & h_d \\ h_d^\top & \hat{h}_d \end{bmatrix}, \tag{C8}$$

where

$$\begin{aligned} \hat{h}_d &= \frac{1}{\sum_{k \in K} \beta_{kd}^{-1}} \\ h_{kd} &= \beta_{kd}^{-1} \hat{h}_d \\ L_{k,k}^d &= \hat{h}_d \beta_{kd}^{-1} \textstyle\sum_{k' \in K \setminus k} \beta_{k'd}^{-1} \\ L_{k,k'}^d &= -\hat{h}_d \beta_{kd}^{-1} \beta_{k'd}^{-1}, \quad k \neq k'. \end{aligned}$$

This results in the solution

$$a_{ikd} = h_{kd} Z_d - Z_{kd} - \sum_{k' \in K} (\alpha_{k'd} - w_{k'd}) L_{k,k'}^d \tag{C9}$$

$$p_{id}^H = \sum_{k \in K} (\alpha_{kd} - w_{kd}) h_{kd} - Z_d \hat{h}_d, \tag{C10}$$

where

$$\begin{aligned} Z_d &= \textstyle\sum_{i \in I, j \in J, k \in K} z_{ijkd} \\ Z_{kd} &= \textstyle\sum_{i \in I, j \in J} z_{ijkd}. \end{aligned}$$

It can be seen that the values of a_{ikd} and P_{id}^H are the same for each firm $i \in I$ and we can drop the index i. Burr Metzler (2000) shows that the arbitrage amounts correspond to the transmission values: $a_{kd} = y_{kd}$.

These substitutions result in the problem formulation

$$\max_{z_{id}} \quad \left(\sum_{k\in K}(\alpha_{kd}-w_{kd})h_{kd}-Z_d\hat{h}_d\right)Z_{id} - \sum_{j\in J,\,k\in K}(\gamma_{ijk}-w_{kd})z_{ijkd} \tag{C11a}$$

$$\text{s.t.} \quad z_{ijkd}\le z^{\max}_{ijkd} \quad (\lambda_{ijkd})\ \forall j\in J,k\in K \tag{C11b}$$

$$z_{ijkd}\ge 0 \tag{C11c}$$

and we can see that the substitutions do not change the concavity of the objective function: the quadratic term for producer $i\in I$ is $\hat{h}_dZ^2_{id}$. Finally, the sales variables s_{ikd} are also eliminated from the market-clearing constraint, resulting in

$$h_{kd}Z_d-Z_{kd}+\sum_{k'\in K}(\alpha_{k'd}-w_{k'd})L^d_{kk'}=y_{kd}\quad (w_{kd})\ \forall k\in K. \tag{C12}$$

The formulation (C11) can be converted into an LCP by using the KKT optimality conditions. The combined KKT conditions of (C11) for all producers $i\in I$ are

$$0\le z_d\perp B_dz_d+\lambda_d+q^z_d\ge 0 \tag{C13a}$$

$$0\le \lambda_d\perp -z_d+z^{\max}_d\ge 0, \tag{C13b}$$

where $q^z_{ijkd}=-\sum_{k\in K}(\alpha_{kd}-w_{kd})h_{kd}+(\gamma_{ijk}-w_{kd})$ and B_d is a PSD matrix with

$$B_d(ijk,i'j'k')=\begin{cases}2\hat{h}_d & i=i'\\ \hat{h}_d & i\ne i',\end{cases}$$

making the bottom level an LCP with a PSD coefficient matrix $\begin{bmatrix}B_d & I\\ -I & 0\end{bmatrix}$. This makes the problem setting suitable for the method described in Section 2.3, but we will continue by presenting the strong duality approach to this problem.

C.1. Strong duality reformulation of the trilevel electricity market model

Using the primal–dual conversion rules for quadratic programs summarised in Dorn (1960), the dual of the bottom-level problem (C11) can be stated as

$$\min_{\lambda_{ijkd},\,z_{ijkd}} \quad \hat{h}_dZ^2_{id}+\sum_{j\in J,\,k\in K}z^{\max}_{ijkd}\lambda_{ijkd} \tag{C14a}$$

$$\text{s.t.} \quad -\lambda_{ijkd}\le \hat{h}d(Z_d+Z_{id})-\textstyle\sum_{k\in K}h_{kd}(\alpha_{kd}-w_{kd}) + (\gamma_{ijk}-w_{kd})\ \ (z_{ijkd})\ \forall j\in J,k\in K \tag{C14b}$$

$$z_{ijkd},\lambda_{ijkd}\ge 0. \tag{C14c}$$

As described in Section 2.4, we impose a *strong duality* constraint stating that the objective value of the dual (minimisation) problem is less or equal to that of the primal (maximisation) problem, and combine constraints (C11b)–(C11c), (C14b)–(C14c) and the strong duality constraint. A solution that satisfies these constraints must be optimal to (C11) and (C14). Notice that the inequality version of the strong duality constraint is convex (as opposed to an equality constraint between the primal and dual objective values), and the other constraints are affine.

Finally, we can write the primal and dual constraints and the strong duality constraint as

$$z_{ijkd}\le z^{\max}_{ijkd}\quad (\lambda'_{ijkd})\ \forall i\in I,j\in J,k\in K \tag{C15a}$$

$$-\lambda_{ijkd}-\hat{h}_d(Z_d+Z_{id})\le -\sum_{k\in K}(\alpha_{kd}-w_{kd})h_{kd} + (\gamma_{ijkd}-w_{kd})\quad (z'_{ijkd})\ \forall i\in I,j\in J,k\in K \tag{C15b}$$

$$\sum_{i\in I}(\hat{h}_dZ^2_{id}+\hat{h}_dZ_dZ_{id}+\sum_{j\in J,\,k\in K}z^{\max}_{ijkd}\lambda_{ijkd} -(\sum_{k\in K}(\alpha_{kd}-w_{kd})h_{kd})Z_{id} + \sum_{j\in J,\,k\in K}(\gamma_{ijk}-w_{kd})z_{ijkd})\le 0\quad (\epsilon_d) \tag{C15c}$$

$$z_{ijkd},\lambda_{ijkd}\ge 0, \tag{C15d}$$

where the strong duality constraints for all producers $i\in I$ have been combined into a single constraint (C15c) to reduce the number of constraints as suggested in Pineda et al. (2018).

The KKT conditions of the ISO problem (13) combined with the constraints (C15) and the market-clearing constraint (C12) are

$$0\le z_{ijkd}\perp \lambda'_{ijkd}-\hat{h}_d(Z_d+Z_{id})+(2\hat{h}_d(Z_d+Z_{id})-\sum_{k\in K}(\alpha_{kd}-w_{kd})h_{kd}+ (\gamma_{ijkd}-w_{kd}))\epsilon_d-(\mathbb{I}(j\in R)-\rho)\psi_{kd}\ge 0 \quad \forall i\in I,j\in J,k\in K \tag{C16a}$$

$$0\le \lambda_{ijkd}\perp -z'_{ijkd}+z^{\max}_{ijkd}\epsilon_d\ge 0\ \forall i\in I,j\in J,k\in K \tag{C16b}$$

$$0\le z'_{ijkd}\perp \lambda_{ijkd}+\hat{h}_d(Z_d+Z_{id})-\sum_{k\in K}(\alpha_{kd}-w_{kd})h_{kd} + (\gamma_{ijkd}-w_{kd})\ge 0\ \forall i\in I,j\in J,k\in K \tag{C16c}$$

$$0\le \lambda'_{ijkd}\perp z^{\max}_{ijkd}-z_{ijkd}\ge 0\ \forall i\in I,j\in J,k\in K \tag{C16d}$$

$$0\le \phi^-_{ad}\perp T^-_a+\sum_{k\in K}PTDF_{ka}y_{kd}\ge 0\ \forall a\in A \tag{C16e}$$

$$0\le \phi^+_{ad}\perp T^+_a-\sum_{k\in K}PTDF_{ka}y_{kd}\ge 0\ \forall a\in A \tag{C16f}$$

$$0\le \psi_{kd}\perp \sum_{i\in I,\,j\in R}z_{ijkd}-\rho\sum_{i\in I,\,j\in J}z_{ijkd}\ge 0\ \forall k\in K \tag{C16g}$$

$$0\le \epsilon_d\perp -\sum_{i\in I}(\hat{h}_dZ^2_{id}+\hat{h}_dZ_DZ_{id}+\sum_{j\in J,\,k\in K}z^{\max}_{ijkd}\lambda_{ijkd}- \left(\sum_{k\in K}(\alpha_{kd}-w_{kd})h_{kd}\right)Z_{id}+\sum_{j\in J,\,k\in K}(\gamma_{ijk}-w_{kd})z_{ijkd})\ge 0 \tag{C16h}$$

$$w_{kd}=\sum_{a\in A}PTDF_{ka}(\phi^+_{ad}-\phi^-_{ad})\ \forall k\in K \tag{C16i}$$

$$y_{kd}=-Z_{kd}+h_{kd}Z_d+\sum_{k'\in K}L^d_{k,k'}(\alpha_{k'd}-w_{k'd})\ \forall k\in K. \tag{C16j}$$

The indicator term $\mathbb{I}(j\in R)$ is 1 if $j\in R$, 0 otherwise, and the variables z'_{ijkd} and λ'_{ijkd} are the dual variables of the primal and dual constraints from the producer level for the system operator problem, and ϵ is the dual variable for the strong duality constraint.

Aalto-DT 158/2024

ISBN 978-952-64-1950-3 (printed)
ISBN 978-952-64-1951-0 (pdf)
ISSN 1799-4934 (printed)
ISSN 1799-4942 (pdf)

Aalto University
School of Science
Department of Mathematics and Systems Analysis
www.aalto.fi

BUSINESS + ECONOMY

ART + DESIGN + ARCHITECTURE

SCIENCE + TECHNOLOGY

CROSSOVER

DOCTORAL THESES