Ein Verfahren zur Dekomposition mehrstufiger stochastischer Optimierungsprobleme mit Ganzzahligkeit und Risikoaversion

Vom Fachbereich Mathematik
der Universität Duisburg-Essen
(Campus Duisburg)
zur Erlangung des akademischen Grades eines
Doktors der Naturwissenschaften
genehmigte Dissertation

von

Thomas Heinze
aus
Wschowa

Referent: Prof. Dr. R. Schultz
Korrefferent: Prof. Dr. W. Römisch

Tag der mündlichen Prüfung: 11.06.2008

Danksagung

An erster Stelle möchte ich mich bei Prof. Dr. Rüdiger Schultz für die Betreuung dieser Arbeit bedanken. Er stand mir jederzeit als Ansprechpartner mit wertvollen Hinweisen zur Verfügung. Sein Engagement und seine Bereitschaft, mich stets zu unterstützen, haben diese Arbeit erst ermöglicht.

Desweiteren möchte ich mich bei allen Mitgliedern der Arbeitsgruppe "Diskrete Mathematik und Optimierung" für die anregenden Diskussionen bedanken. Insbesondere danke ich Dr. Ralf Gollmer, der mir bei allen EDV-Problemen stets hilfreich zur Seite stand.

Mein Dank gilt Prof. Dr. Werner Römisch für seine Bereitschaft als zweiter Referent dieser Arbeit zur Verfügung zu stehen.

Bei der Deutschen Forschungsgemeinschaft bedanke ich mich für die finanzielle Unterstützung des Projekts.

Inhaltsverzeichnis

Kapitel 1

Einleitung

Viele Entscheidungen, deren Ausgang von zukünftigen Ereignissen abhängt, müssen unter Unsicherheit getroffen werden. Man denke beispielsweise an quantitative Produktionsentscheidungen, denen eine zufallsabhängige Nachfrage zugrunde liegt oder an Investitionsentscheidungen, deren Ausgang von den zufälligen Schwankungen der Finanzmärkte beeinflusst wird. Die stochastische Optimierung entwickelt mathematische Modelle, um für solche Probleme eine möglichst optimale Entscheidung zu treffen. Die Schwierigkeiten sind sowohl konzeptionell beim Entwerfen geeigneter Modelle, als auch numerisch beim Lösen konkreter mathematischer Optimierungsprobleme.

In dieser Arbeit wird das Kompensations- oder mehrstufige Modell betrachtet, welches auf stochastische, lineare, gemischt-ganzzahlige Optimierungsprobleme angewendet wird, siehe [18, 54, 59, 60, 91, 100]. Diese Problemklasse stellt innerhalb der stochastischen Optimierung eine der am meist genutzten Planungsinstrumente in der Praxis dar. Beispiele hierfür gibt es in den Bereichen Energieversorgung, Telekommunikation, Produktion, Logistik, Finanzplanung und Lagerhaltung, vergleiche auch [117].

Ein lineares, gemischt-ganzzahliges Optimierungsproblem ist gegeben durch

$$\inf_x \{c^\top x : Ax \leq b,\, x \in X\}, \tag{1.1}$$

mit den Daten $c \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, $A \in \mathbb{R}^{n \times m}$ und einer nicht-leeren, polyedrischen

Menge $X \subseteq \mathbb{Z}^{n'} \times \mathbb{R}^{n''}$, welche Ganzzahligkeitsanforderungen an x induziert. Wir werden in dieser Arbeit den Kontext der Kostenminimierung verwenden.

Die Lineare Optimierung, siehe auch [24, 87, 104], spielt seit der Entwicklung des Simplex-Algorithmus von G. Dantzig [26] eine wichtige Rolle. Sie gehört zum Gebiet der konvexen Optimierung und es existieren sowohl theoretisch als auch praktisch effiziente Algorithmen.

Wie in anderen Bereichen der mathematischen Optimierung ändern Ganzzahligkeitsanforderungen die Eigenschaft der Probleme signifikant. Im Unterschied zu linearen Problemen ohne Ganzzahligkeit, welche mit Innere-Punkte-Verfahren in polynomialer Zeit lösbar sind, sind ganzzahlige, lineare Probleme im Allgemeinen NP-schwer. Dadurch ist die algorithmische Behandlung weit schwieriger und die zurzeit berechenbaren Problemgrößen entsprechend kleiner, siehe [81, 88].

Da die Daten in (1.1) bekannt sind, spricht man auch von einem deterministischen Problem. Sind diese abhängig von einem zufälligen Parameter $\omega \in \Omega$, mit einer Ereignismenge Ω, erhält man ein stochastisches, lineares, gemischt-ganzzahliges Optimierungsproblem.

$$\inf_x \{c(\omega)^\top x : A(\omega)x \leq b(\omega),\, x \in X\} \tag{1.2}$$

Anders als beispielsweise in der Robusten Optimierung in [16], setzen wir voraus, dass wir weitere Informationen über die Ereignismenge Ω in Form einer Wahrscheinlichkeitsverteilung besitzen. Das heißt, die Daten c, b und A sollen auf einem gegebenen Wahrscheinlichkeitsraum $(\Omega, \mathcal{F}, \mathbb{P})$ definierte Zufallsvariablen sein.

Durch die Abhängigkeit von ω ist weder die Zielfunktion noch der zulässige Bereich in (1.2) eindeutig bestimmt. Um diesem Problem dennoch einen optimalen Wert zuzuordnen, sind verschiedene Modelle mit unterschiedlichen Philosophien entstanden. Wir werden wie bereits erwähnt das Kompensationsmodell anwenden. In der Literatur findet man eine Vielzahl anderer Modelle wie beispielsweise Wahrscheinlichkeitsrestriktionen, dessen Ansatz auf A. Charnes und W.W. Cooper zurückgeht, siehe [21, 22, 23, 91] oder die Robuste Optimierung, siehe [15, 16]. Weitere Modellansätze findet man unter anderem in [4, 18, 60, 70, 75].
Das Kompensationsmodell wurde eingeführt in [12] und [27]. In seiner ursprünglichsten Form unterscheidet man Variablen der ersten Stufe x und zweiten Stufe y. Es

wird der folgende zeitliche Ablauf zugrunde gelegt:

$$\text{Entscheide } x \longrightarrow \text{beobachte die Realisierung in } \Omega \longrightarrow \text{entscheide } y. \tag{1.3}$$

Die in dieser Festlegung versteckte Informationsrestriktion -x muss ohne Kenntnis der Realisierung der zufälligen Ereignisse festgelegt werden- wird als Nichtantizipativitätsbedingung bezeichnet. Die zufallsabhängigen Größen werden nicht durch die Erststufenentscheidung beeinflusst. Gemäß dieses Modellierungsrahmens betrachten wir folgendes Problem

$$\inf_x \{c(\omega)^\top x + \Psi(x,\omega) : x \in X\} \tag{1.4}$$

mit

$$\Psi(x,\omega) = \inf_y \{q^\top y : \ A(\omega)x + Wy = b(\omega),\ y \in Y\}. \tag{1.5}$$

Hier haben wiederum alle Inputdaten Dimensionen entsprechender Größe und X, Y sind wiederum polyedrische Mengen, die Ganzzahligkeitsanforderungen an x bzw. y induzieren. Wie bei jedem mathematischen Modell, muss bei einer praktischen Anwendung überprüft werden, ob die Nutzung des Modells sinnvoll ist. Wie der Name schon sagt, muss zumindest bis zu einem gewissen Grad die Möglichkeit der Kompensation, die hier durch die Zweitstufenvariable verkörpert wird, gegeben sein. Falls die Kompensation keine zusätzlichen Bedingungen an die Erststufenentscheidung induziert, spricht man von vollständiger Kompensation. Sie ist gegeben falls

$$\forall t \in \mathbb{R}^m \quad \exists y \in Y \ : \ Wy = t. \tag{1.6}$$

Es sei bemerkt, dass (1.4) ohne weitere Voraussetzungen keinesfalls wohldefiniert ist. Insbesondere muss ein Entscheidungskriterium für die Zielfunktion festgelegt werden. Setzen wir voraus, dass

$$\Psi : \mathbb{R}^n \times \Omega \longrightarrow \mathbb{R} \tag{1.7}$$

messbar ist, so kann die Menge $\{c(\omega)^\top x + \Psi(x,\omega)\}_{x\in X}$ als Familie von Zufallsvariablen und das Problem (1.4) als Suche nach der "besten" Zufallsvariable dieser Familie verstanden werden. Somit können statistische Parameter als Entscheidungskriterien gewählt werden. Im klassischen Fall greift man auf den Erwartungswert zurück, so dass folgendes Optimierungsproblem entsteht:

$$\inf_x \{\mathbb{E}[c(\omega)^\top x + \Psi(x,\omega)] : x \in X\}. \tag{1.8}$$

In dieser Arbeit werden wir uns nicht auf den zweistufigen Fall beschränken, sondern das mehrstufige Modell betrachten, das einen beliebigen, diskreten Zeithorizont zulässt. Als einführende Lektüre sei auf [97] verwiesen. Im mehrstufigen Modell findet eine Erweiterung der Interaktion in (1.3) zwischen Entscheidung und Beobachtung statt. Aufgrund der daraus resultierenden komplexeren Dynamik nimmt die angesprochene Nichtantizipativitätsbedingung eine größere Rolle ein als im zweistufigen Fall. Diese Tatsache macht das mehrstufige Modell aus struktureller und algorithmischer Sicht zu einem wesentlich komplexeren Objekt.

Risikoaversion wird bei der Entscheidungsfindung zunehmend berücksichtigt. Wie Risiko genau zu definieren und in stochastische Optimierungsprobleme einzubinden ist, ist ein weites Gebiet intensiver Forschung. Zu den wohl bekanntesten Beiträgen gehören das Mean-Risk Modell von H.M. Markowitz [73], die Entwicklung der kohärenten Risikomaße in P. Artzner et al. [8] und das Prinzip der stochastischen Dominanz, siehe die Arbeiten von H. Levy [68] und A. Müller und D. Stoyan [79]. Wir werden in dieser Arbeit das Risiko mithilfe von Risikomaßen berücksichtigen. Diese werden gemäß dem von H.M. Markowitz entwickelten Mean-Risk Modell als zusätzliches Entscheidungskriterium neben dem Erwartungswert in (1.8) mit einfließen.

Es ist bekannt, dass bei diskreter Wahrscheinlichkeitsverteilung zwei- und mehrstufige Modelle zu großen, gemischt-ganzzahligen Optimierungsproblemen werden. Aufgrund der Größe dieser Probleme sind sie mit Standardprogrammen wie Xpress-MP [28] oder CPLEX [57] nur schwer bis gar nicht lösbar. Der Schwerpunkt dieser Arbeit liegt in der numerischen Behandlung, der durch die Modellbildung entstehenden mathematischen Optimierungsprobleme bei diskreten Wahrscheinlichkeitsverteilungen. Im risikoneutralen Fall weisen diese Optimierungsprobleme eine blockstrukturierte Nebenbedingungsmatrix auf. Wir werden eine Klasse von Risikomaßen für das Mean-Risk Modell nutzen, die diese Eigenschaft erhält. Dies sind die in A. Eichhorn und W. Römisch [34] definierten polyedrischen Risikomaße. Diese Blockstruktur erlaubt die Anwendung von Dekompositionsverfahren, siehe hierzu beispielsweise [99]. Wir werden ein Verfahren entwickeln, das an die Arbeiten von Alonso et al. [5, 6] angelehnt ist und auf einem Branch-&-Bound Schema basiert, siehe auch [50]. In

[5, 6] werden im Gegensatz zu unseren Problemen, mehrstufige, risikoneutrale Modelle im rein binären Fall betrachtet.

Die Arbeit ist wie folgt aufgebaut: In Kapitel 2 wird im ersten Abschnitt das mehrstufige, stochastische, risikoneutrale Modell vorgestellt. Wir werden dann im zweiten Abschnitt auf die Risikomodellbildung eingehen. Zuerst werden einige Risikomaße für das Mean-Risk Modell anhand einer sehr allgemeinen Klasse von Zufallsvariablen vorgestellt. Danach werden die Auswirkungen der Modellerweiterung auf mehrstufige Kompensationsmodelle untersucht.

In Kapitel 3 werden zuerst unsere Modelle im Fall einer diskreten Wahrscheinlichkeitsverteilung betrachtet. Wir werden dann im Abschnitt 3.3 ein allgemeines Lösungsschema vorstellen, welches in den nächsten Unterabschnitten zu den konkreten Algorithmen bb-tree und bb-hash weiterentwickelt wird.

In Kapitel 4 werden zwei Anwendungen vorgestellt, an denen sowohl konzeptionelle Aspekte des Modells, als auch die Rechengeschwindigkeit der Algorithmen veranschaulicht werden soll. Zusätzlich wird ein interner Vergleich der Algorithmen bb-tree und bb-hash angestellt.

Kapitel 2

Mehrstufige Kompensationsmodelle

2.1 Allgemeines

Im zweistufigen Modell geht man davon aus, dass die zufallsbehafteten Informationen zu einem festen Zeitpunkt vollständig bekannt werden. Bei vielen realen Problemen erhält man die Kenntnis über diese Informationen erst allmählich im Laufe der Zeit. Um diesem Sachverhalt Rechnung zu tragen, wird das in der Einleitung vorgestellte zweistufige Modell (1.8) zu einem mehrstufigen Modell erweitert.

Es hat sich gezeigt, dass insbesondere die Gannzahligkeitsanforderungen erheblichen Einfluß auf die Struktur der entstehenden Optimierungsprobleme haben. Im rein linearen Fall entstehen bei zwei- und mehrstufigen, risikoneutralen Modellen, stetige, konvexe Optimierungsprobleme, siehe etwa [100], so dass das breite Wissen insbesondere der konvexen Analysis [55, 92] hier angewendet werden kann. Diese beiden, in der Optimierung so wichtigen Eigenschaften, gehen bei Ganzahligkeitsanforderungen im Allgemeinen verloren. Aus diesem Grund ist im Vergleich zum rein linearen Fall nur wenig über die Struktur dieser Probleme bekannt.

Im Gegensatz zum mehrstufigen, gemischt-ganzzahligen Modell, wurde das zweistufige Modell über einen längeren Zeitraum erforscht, so dass das Verständnis entsprechend größer ist. Siehe hierzu für den risikoneutralen Fall die Übersichtsartikel von W.K. Klein Haneveld und M.H. van der Vlerk [64] und F.V. Louveaux und R.

Schultz [69]. Die Struktur zweistufiger Mean-Rsik Modelle wurde unter anderem in [72, 109, 110, 115] untersucht.

Mehrstufige Modelle weisen durch den dynamischen Ansatz eine weit höhere Komplexität auf. Diese wird insbesondere durch den Übergang von einer einzelnen Zufallsvariablen zu einem mehrstufigen stochastischen Prozess bei der Datenmodellierung verursacht. Es ist nur sehr wenig über die Struktur mehrstufiger Erwartungswert- sowie mehrstufiger Mean-Risk Modelle bekannt, es sei auf [34, 97] verwiesen.

Durch den Übergang zu einem stochastischen Prozess, wird auch eine adäquate Berücksichtigung des Risikos auf der Modellierungsebene schwieriger. So sollte beispielsweise nicht nur das Auskommen in der finalen Zeitstufe, sondern auch das Auskommen in den einzelnen Zwischenstufen, sowie die Interaktion zwischen den Zeitstufen als Entscheidungskriterium mit einbezogen werden. Zum Thema mehrstufiges Risiko siehe die Arbeiten von P. Artzner et al. [9, 10], S. Weber [119], A. Ruszczyński und A. Shapiro [102] und das kürzlich erschiene Buch von G. CH. Pflug und W. Römisch [90].

Das Kapitel weist folgenden Aufbau auf: Wir werden im ersten Abschnitt das erwartungswertbasierte mehrstufige Modell vorstellen.

Im zweiten Abschnitt wird auf das Thema Risikominimierung genauer eingegangen. Im Unterabschnitt 2.3.1 werden das Mean-Risk Modell und einige dafür nötige Risikomaße vorgestellt.

Im zweiten Unterabschnitt 2.3.2 wird eine Klasse von Risikomaßen betrachtet, die polyedrischen Risikomaße, mit denen wir unser Erwartungswertmodell zu einem Mean-Risk Modell erweitern werden.

2.2 Erwartungswertmodell

Wir betrachten ein dynamisches System, das in $T \in \mathbb{N}$ verschiedenen diskreten Stufen beobachtbar ist, siehe Abbildung 2.1.

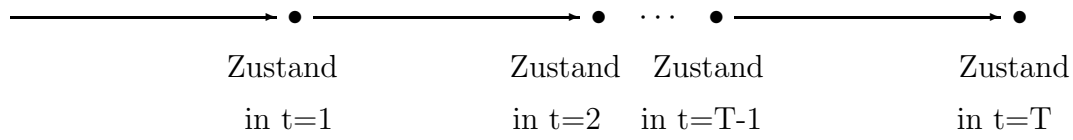


Abb. 2.1: Dynamischer Verlauf

Daran angelehnt definieren wir das folgende lineare, gemischt-ganzzahlige, dynamische Problem:

$$\min \left\{ \sum_{t=1}^{T} c_t^\top x_t \;\middle|\; \begin{array}{l} x_t \in X_t,\ t = 1, \ldots, T \\ \sum\limits_{\tau=1}^{t} A_{t\tau} x_\tau \leq b_t,\ t = 1, \ldots, T \end{array} \right\}, \tag{2.1}$$

mit den Daten $c_t \in \mathbb{R}^{n_t}$, $b_t \in \mathbb{R}^{m_t}$, $A_{t\tau} \in \mathbb{R}^{m_t \times n_\tau}$, $\tau = 1, \ldots, t$ und nicht-leeren, polyedrischen Mengen $X_t \subseteq \mathbb{Z}^{n_t'} \times \mathbb{R}^{n_t''}$, $t = 1, \ldots, T$. Diese Daten sind den einzelnen Stufen zugeordnet. Mit anderen Worten, legt der Datensatz c_t, b_t, $A_{t\tau}$, $\tau = 1, \ldots, t$ und X_t den Zustand des Systems in $t \in \{1, \ldots, T\}$ fest. Da die Entscheidung x_t in einer Stufe $t \in \{1, \ldots, T\}$ die Entscheidung $x_{t'}$ in allen Zeitstufen $t' \in \{1, \ldots, T\}$ mit $t' \geq t$ beinflusst, induzieren wir eine sequentielle Entscheidungsfindung von $t = 1$ bis $t = T$.

Wir setzen nun voraus, dass das Verhalten des Systems zwischen den einzelnen Stufen zufallsabhängig ist. Das heißt, die obigen Datensätze zu einem Zeitpunkt $1 \leq t \leq T$ sind abhängig von einer zufälligen Größe. Modelliert wird diese Zufallsabhängigkeit des Systems mithilfe eines stochastischen Prozesses $\{\xi_t\}_{t=1}^T$ mit Werten im $\times_{t=1}^T \mathbb{R}^{s_t}$, der auf einem Wahrscheinlichkeitsraum $(\Omega, \mathcal{F}, \mathbb{P})$ definiert ist. Es werden keine zusätzlichen Einschränkungen gemacht, inwieweit sich die zufälligen Ereignisse in den einzelnen Stufen beeinflussen, d.h. zwei Zufallsvariablen ξ_t und $\xi_{t'}$ mit $t, t' \in \{1, \ldots, T\}$, $t \neq t'$ können voneinander abhängig sein. Wie im zweistufigen Modell, ist eine Abhängigkeit der zufälligen Ereignisse von den Entscheidungen nicht zugelassen. Nimmt man wiederum den Erwartungswert als Entscheidungskriterium, geht das Problem (2.1) über in

$$\min \left\{ \mathbb{E}\left[\sum_{t=1}^{T} c_t(\xi_t)^\top x_t \right] \;\middle|\; \begin{array}{l} x_t \in X_t,\ t = 1, \dots, T \\ \sum\limits_{\tau=1}^{t} A_{t\tau}(\xi_t) x_\tau \leq b_t(\xi_t),\ t = 1, \dots, T \end{array} \right\}, \tag{2.2}$$

wobei wir annehmen dass die Menge X_t nicht zufallsabhängig ist und die Daten $c_t(\cdot), b_t(\cdot)$ und $A_{t\tau}(\cdot)$ in (2.2) affin lineare Funktionen der entsprechenden Komponenten von ξ, für alle $\tau = 1, \dots, t$, $t = 1, \dots, T$ sind. Um sicherzustellen, dass der Erwartungswert existiert, muss das Skalarprodukt $c_t(\xi_t)^\top x_t$ integrierbar sein. Die Integrierbarkeit ist nach der Hölderschen Ungleichung gegeben, falls das Produkt $\|\xi_t\| \|x_t\|$ integrierbar ist, was wiederum durch die Bedingungen $\xi_t \in L_{p_t}(\Omega, \mathcal{F}_t, \mathbb{P}; \mathbb{R}^{s_t})$ und $x_t \in L_{q_t}(\Omega, \mathcal{F}_t, \mathbb{P}; \mathbb{R}^{n_t})$, mit $p_t, q_t \in [1, \infty]$ und $\frac{1}{p_t} + \frac{1}{q_t} = 1$ gegeben ist. Um an den stochastischen Prozess ξ so wenig Bedingungen wie nötig zu stellen, sei im Weiteren vorausgesetzt, dass $\xi_t \in L_1(\Omega, \mathcal{F}_t, \mathbb{P}; \mathbb{R}^{s_t})$ und $x_t \in L_\infty(\Omega, \mathcal{F}_t, \mathbb{P}; \mathbb{R}^{n_t})$. Das Problem (2.2) ist also als Optimierungsproblem über $\times_{t=1}^{T} L_\infty(\Omega, \mathcal{F}_t, \mathbb{P}; \mathbb{R}^{n_t})$ zu verstehen, in dem die Restriktionen punktweise $\mathbb{P}$-fast sicher gelten sollen.

Da bei der Entscheidung in einer beliebigen Stufe $1 \leq t \leq T$ die gesamten Informationen des zufälligen Verlaufs in (2.2) genutzt werden, spricht man von einer Wait-and-See Entscheidung, vergleiche [18, 61, 91, 100]. Die Entscheidungsfindung wird in die letzte Stufe T verlegt, wie Abbildung 2.2 veranschaulicht.

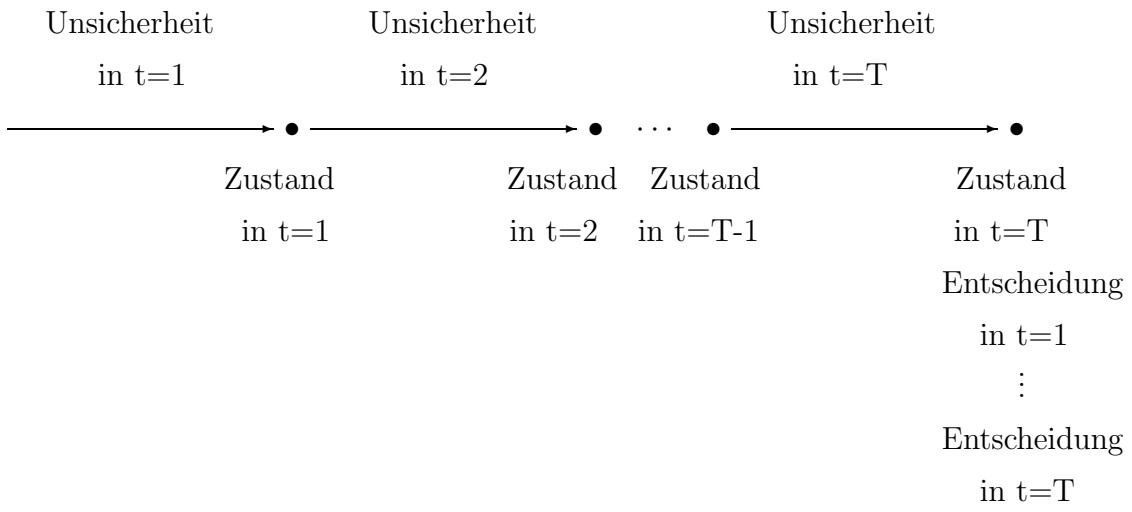


Abb. 2.2: Wait-and-See Entscheidung

Das Kompensationsmodell verlangt eine Folge nichantizipativer- bzw. Here-and-Now Entscheidungen, wie in (1.3). Dazu betrachten wir σ-Algebren $\mathcal{F}_t := \sigma(\xi_1, \ldots, \xi_t)$, die durch die zufälligen Vektoren $(\xi_1, \ldots, \xi_t)$ erzeugt werden. Offenbar gilt $\mathcal{F}_t \subseteq \mathcal{F}_{t+1}$, $t = 1, \ldots, T-1$, da die verfügbaren Informationen im Laufe der Zeit zunehmen und sich die σ-Algebren dadurch verfeinern. Es sei ohne Beschränkung der Allgemeinheit vorausgesetzt, dass die Entscheidungen in der ersten Stufe zufallsunabhängig getroffen werden, das heißt $\mathcal{F}_1 = \{\emptyset, \Omega\}$. Außerdem soll gelten, dass in der letzten Stufe alle Ereignisse bekannt sind, $\mathcal{F}_T = \mathcal{F}$. Um nun zu garantieren, dass in dem mathematischen Optimierungsproblem nur die Informationen verwendet werden, die bis zu einem bestimmten Zeitpunkt $t \in \{1, \ldots, T\}$ zur Verfügung stehen, muss die Entscheidungsvariable x_t zu diesem Zeitpunkt $\mathcal{F}_t$-messbar sein. Diese Meßbarkeitsbedingungen können mithilfe des bedingten Erwartungswertes dargestellt werden durch $H_t(x_t) := x_t - \mathbb{E}[x_t|\mathcal{F}_t] = 0$, $t = 1, \ldots, T$, siehe beispielsweise [11]. Mit diesen zusätzlichen Bedingungen geht (2.2) über in

$$\min \left\{ \mathbb{E}\left[\sum_{t=1}^{T} c_t(\xi_t)^\top x_t \right] \;\middle|\; \begin{array}{l} H_t(x_t) = 0,\ t = 1, \ldots, T \\ x_t \in X_t,\ t = 1, \ldots, T \\ \sum_{\tau=1}^{t} A_{t\tau}(\xi_t) x_\tau \leq b_t(\xi_t),\ t = 1, \ldots, T \end{array} \right\}. \tag{2.3}$$

Das Problem (2.3) ist die mehrstufige Erweiterung des zweistufigen Modells (1.8) welches den gewünschten Entscheidungsverlauf in Abbildung 2.3 hat.

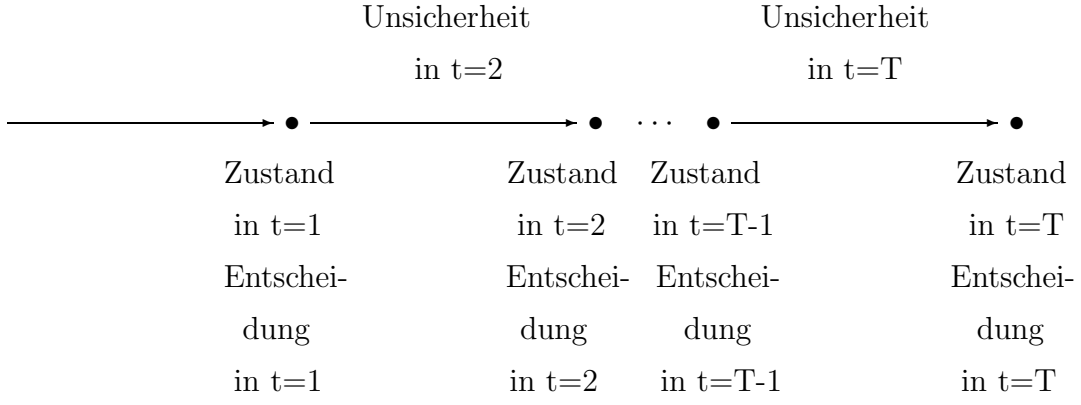


Abb. 2.3: Nichantizipative Entscheidungen

Bemerkung 2.1 *i) Durch die Nichtantizipativitätsbedingungen wird ein linearer Untervektorraum im* $\times_{t=1}^{T} L_\infty(\Omega, \mathcal{F}_t, \mathbb{P}; \mathbb{R}^{n_t})$ *erzeugt.*
ii) Die Nichtantizipativitätsbedingungen sind im zweistufigem Modell (1.8) implizit durch die Unabhängigkeit der Variablen x *von* ω *in (1.5) gegeben.*
iii) Die mehrstufige Erweiterung der Formulierung in (1.8) führt zu einer verschachtelten Folge bedingter Erwartungswerte:

$$\min\{\Phi(x_1) := \mathbb{E}[c_1(\xi_1)^\top x_1 + \Psi(x_1, \xi)] : x_1 \in X_1,\ A_{11}x_1 = b_1\} \tag{2.4}$$

$$= \min_{\substack{x_1 \in X_1 \\ A_{11}x_1 = b_1}} \{c_1^\top x_1$$

$$+\mathbb{E}_{\mathcal{F}_2}[\min_{\substack{x_2 \in X_2 \\ A_{21}(\xi_1)x_1 + A_{22}(\xi_2)x_2 = b_2(\xi_2)}} \{c_2(\xi_2)^\top x_2 + \dots$$

$$+\mathbb{E}_{\mathcal{F}_{T-1}}[\min_{\substack{x_{T-1} \in X_{T-1} \\ A_{T-1,1}(\xi_1)x_1 + \dots + A_{T-1,T-1}(\xi_{T-1})x_{T-1} = b_{T-1}(\xi_{T-1})}} \{c_{T-1}(\xi_{T-1})^\top x_{T-1}$$

$$+\mathbb{E}_{\mathcal{F}_T}[\min_{\substack{x_T \in X_T \\ A_{T,1}(\xi_1)x_1 + \dots + A_{T,T}(\xi_T)x_T = b_T(\xi_T)}} \{c_T(\xi_T)^\top x_T\}]\}] \dots\}]\},$$

wobei $\mathbb{E}_{\mathcal{F}_t}[\,\cdot\,]$, $t = 1, \dots, T$ *den bedingten Erwartungswert* $\mathbb{E}[\,\cdot\,|\mathcal{F}_t]$ *darstellt. Eine ausführliche Herleitung dieser Darstellung findet man unter anderem in Kapitel 2.4 in [108].*

Zum besseren Verständnis soll noch eine weitere Sichtweise auf Problem (2.3) gegeben werden. Es sei $\mathcal{X}(\xi)$ die Menge aller $x \in \times_{t=1}^{T} L_\infty(\Omega, \mathcal{F}, \mathbb{P}; \mathbb{R}^{n_t})$, die die Nebenbedingungen in (2.3) erfüllen. Dann induziert jedes zulässige $x \in \mathcal{X}(\xi)$ Kostenvariablen

$$z_t := \sum_{\tau=1}^{t} c_\tau(\xi_\tau)^\top x_\tau,\ \ t = 1, \dots, T. \tag{2.5}$$

Analog zum zweistufigen Fall kann (2.3) wiederum als Suche nach der "besten" dieser Kostenvariablen interpretiert werden, wobei man im mehrstufigen Fall ent-

scheiden muss, ob nur die finalen Kosten z_T von Interesse sind, oder ob die Wahl nach der "besten" Kostenvariablen den gesamten Prozess begleitet. Nimmt man den Erwartungswert als Entscheidungskriterium, so kann (2.3) dargestellt werden durch

$$\min \{\mathbb{E}[z_T] \,|x \in \mathcal{X}(\xi)\}. \tag{2.6}$$

Wir werden (2.5) im Weiteren als akkumulierten Kostenprozess bezeichnen, in dem für jedes $1 \leq t \leq T$ die entstehenden Kosten bis zur Stufe t aufsummiert werden. Sind die Kosten in den einzelnen Zeitschritten von Interesse, so wird man den Kostenprozess folgendermaßen definieren

$$\tilde{z}_t := c_t(\xi_t)^\top x_t, \quad t = 1, \ldots, T. \tag{2.7}$$

Siehe hierzu auch die Diskussion in [34] mit den entsprechenden Referenzen. Beide Prozesse lassen sich problemlos ineinander überführen. So ist

$$\tilde{z}_t = z_{t+1} - z_t, \quad t = 1 \ldots, T-1 \tag{2.8}$$

und

$$z_t = \sum_{\tau=1}^{t} \tilde{z}_\tau. \tag{2.9}$$

2.3 Risikoaversion

In obigen Modellen wurde der Erwartungswert als alleiniges Entscheidungskriterium für die Wahl der 'besten' Zufallsvariablen in Betracht gezogen. Dieser berücksichtigt nicht die Streuung der zufällig anfallenden Kosten. Die Frage, wie die Streuung bzw. das Risiko in die Entscheidungsfindung mit einzubeziehen ist, ist Gegenstand intensiver Forschung, siehe [8, 30, 38, 73, 79, 103].

Wir werden in dieser Arbeit das Risiko mithilfe von Risikomaßen berücksichtigen. Eine andere Möglichkeit risikoaverse Modelle zu formulieren, ist beispielsweise die Anwendung von Dominanzrestriktionen, siehe hierzu [30, 31, 41, 42, 43]. Wir benutzen die folgenden Notationen. Es sei $\mathcal{Z}$ die Menge aller reellen Zufallsvariablen

auf einem gegebenen Wahrscheinlichkeitsraum $(\Omega, \mathcal{F}, \mathbb{P})$. Zu einer Zufallsvariablen $X \in \mathcal{Z}$ ist $F_X(x)$ die zugehörige Verteilungsfunktion. Ein Risikomaß ist ein Funktional

$$\mathcal{R} : \mathcal{Z} \longrightarrow \mathbb{R}. \tag{2.10}$$

Eine Zufallsvariable $X \in \mathcal{Z}$ wird einer Zufallsvariablen $Y \in \mathcal{Z}$ vorgezogen, falls

$$\mathcal{R}(X) < \mathcal{R}(Y). \tag{2.11}$$

Obige Definition lässt weiten Spielraum bei einer konkreten Festlegung von $\mathcal{R}$. Dies ist dem Umstand gewidmet, dass Risikoaversion höchst subjektiv ist. Jeder Entscheidungsträger mag seine eigene Vorstellung haben, wie $\mathcal{R}$ sinnvoll für den Rahmen, in dem er sich bewegt, zu definieren ist.

Risikomaße ändern oft die Eigenschaften der Optimierungsprobleme, so dass beispielsweise die algorithmischen Methoden im risikoneutralen Fall nicht länger angewendet werden können. Wir werden deshalb im nächsten Unterabschnitt im Zusammenhang mit dem Mean-Risk Modell, einige "populäre" Risikomaße vorstellen und sie unter den folgenden Gesichtspunkten betrachten:

1) Erfüllt das Risikomaß die gewünschte Risikoaversion (Modellierungsebene)

2) Welchen Einfluss hat es auf die algorithmische Behandlung des Optimierungsproblems (Algorithmische Ebene)

Wir werden auf den ersten Punkt nur kurz am Ende des Abschnitts 2.3.1 eingehen. Da in dieser Arbeit der Schwerpunkt auf der algorithmischen Behandlung liegt, ist der gesamte Abschnitt 2.3.2 dem letzten Punkt gewidmet. Wir werden dort eine Klasse von Risikomaßen vorstellen, welche sich aus algorithmischer Sicht als sehr nützlich erwiesen hat.

2.3.1 Mean-Risk Modelle

Da Erwartungswert und Risikofunktionale für sich allein nur einen begrenzten Informationsgehalt haben, werden bei der Optimierung oft so genannte Mean-Risk-Modelle betrachtet, deren Ursprung auf H.M. Markowitz [73] zurückgeht. Es handelt

sich um ein bikriterielles Optimierungsproblem aus Erwartungswert und Risikomaß:

$$\min\{(\mathbb{E}[X], \mathcal{R}[X]) : X \in \tilde{\mathcal{Z}}\}, \tag{2.12}$$

mit dem Restriktionsbereich $\tilde{\mathcal{Z}} \subseteq \mathcal{Z}$. Die Vektor- oder Multikriterielle Optimierung ist ein eigenständiges Teilgebiet der mathematischen Optimierung, siehe etwa [33, 65, 80, 114]. Ein allgemein akzeptierter Begriff für die Optimalität dieser Probleme ist der der Effizienz.

Definition 2.2 (Effiziente Lösung, nicht dominierte Punkte)
Eine Lösung $X \in \tilde{\mathcal{Z}}$ heißt effizient, falls es keine andere Lösung $Y \in \tilde{\mathcal{Z}}$ gibt, so dass mindestens eine der beiden Beziehungen erfüllt ist:

i) $\mathbb{E}[Y] < \mathbb{E}[X]$ *und* $\mathcal{R}[Y] \leq \mathcal{R}[X]$

ii) $\mathbb{E}[Y] \leq \mathbb{E}[X]$ *und* $\mathcal{R}[Y] < \mathcal{R}[X]$.

Der Zielfunktionsvektor $(\mathbb{E}[X], \mathcal{R}[X])$ *einer effizienten Lösung $X \in \tilde{\mathcal{Z}}$ heißt nicht-dominierter Punkt.*

Man betrachte Abbildung 2.5, alle gekennzeichneten Punkte sind nicht-dominiert. Es gibt eine Vielzahl von Methoden, wie effiziente Punkte berechnet werden können, siehe beispielsweise [80, 114]. Eine davon ist die Suche nach effizienten Punkten mittels einer gewichteten Summe:

$$\min\{\mathbb{E}[X] + \beta\, \mathcal{R}[X] : X \in \tilde{\mathcal{Z}}\}, \quad \beta \in \mathbb{R}_+. \tag{2.13}$$

In [65] wird gezeigt, dass der Lösungsvektor dieses Optimierungsproblems zu jedem $\beta > 0$ ein nicht-dominierter Punkt ist. Wie einfache Beispiele zeigen, gilt die Umkehrung im Allgemeinen nicht, falls die Funktionen $X \to \mathbb{E}[X]$ und $X \to \mathcal{R}[X]$ nicht konvex sind. In dem Fall existiert für einen nicht-dominierten Punkt $(\mathbb{E}[X_0], \mathcal{R}[X_0])$ der im Inneren der Menge $conv(\{(\mathbb{E}[X], \mathcal{R}[X]) : X \in \tilde{\mathcal{Z}}\})$ liegt kein $\beta > 0$, so dass $(\mathbb{E}[X_0], \mathcal{R}[X_0])$ Lösung von (2.13) ist. Vergleiche P_3 in Abbildung 2.5.

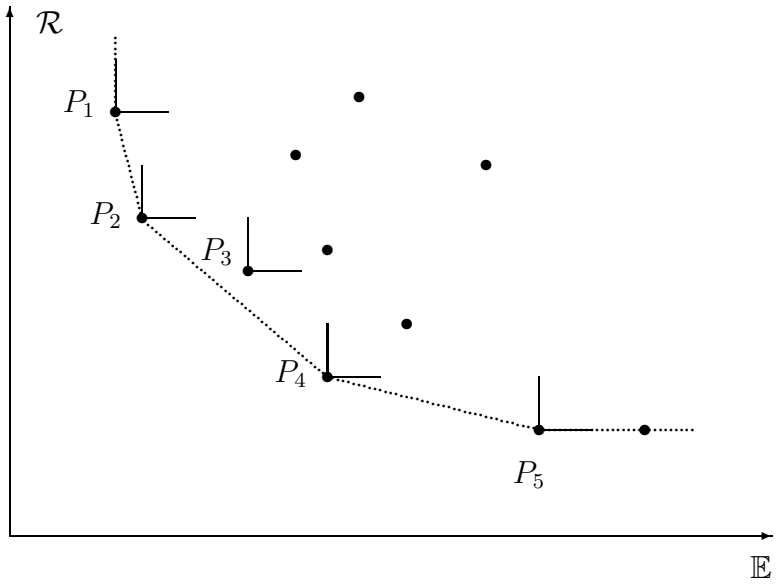


Abb. 2.5: Nicht-dominierte Punkte

Wir werden in unseren Mean-Risk Modellen mit einer gewichteten Summe wie in (2.13) arbeiten. Die Wahl des Parameters β hängt von der Risikoaversion des Entscheidungsträgers ab.

Im Folgenden werden einige Risikomaße vorgestellt, dabei sei vorausgesetzt, dass die Zufallsvariablen die erforderlichen Integrationseigenschaften haben.

Beispiel 2.3 (Varianz)
Eines der populärsten Risikomaße in diesem Zusammenhang ist die Varianz

$$\mathcal{R}_{Var}(X) := \mathbb{E}[(X - \mathbb{E}[X])^2]. \tag{2.14}$$

Die Varianz gehört zu einer Standardgröße der Wahrscheinlichkeitstheorie und wurde schon früh in der Risikominimierung verwendet, siehe etwa [74]. Die Eigenschaften dieses Funktionals sind unter anderem nachzuschlagen in [11].

Beispiel 2.4 (Value-at-Risk)
Der Value-at-Risk berechnet "den kleinsten Wert, der mit Wahrscheinlichkeit (1-α) · 100% höchsten Realisierungen, die X annimmt". Hier ist $\alpha \in (0,1)$ eine vorgegebene Wahrscheinlichkeit.

$$\mathcal{R}_{VaR_\alpha}(X) := \min_{\eta \in \mathbb{R}} \{\eta : \mathbb{P}(X > \eta) \leq 1 - \alpha\}. \tag{2.15}$$

In Abbildung 2.4 wird veranschaulicht, wie die η mit den gewählten α zusammenhängen. Das zu einem gegebenen α optimale η ist mit η_α gekennzeichnet - auf die Bezeichnung η_α^+ wird im nächsten Beispiel eingegangen. Aufgrund der Monotonie und der rechtsseitigen Stetigkeit der Verteilungsfunktion, ist die Existenz des Minimums in (2.15) immer gegeben. Der Value-at-Risk gehört bei der Absicherung finanzieller Risiken zu einem der weit verbreitesten Risikomaße, siehe auch [58].

Beispiel 2.5 (Conditional Value-at-Risk)
Es sei ein $\alpha \in (0,1)$ gegeben. Dann ist der α-Conditional Value-at-Risk definiert durch

$$\mathcal{R}_{CVaR_\alpha} := \inf_{\eta\in\mathbb{R}}\{\eta + \frac{1}{1-\alpha}\mathbb{E}[(X-\eta)^+]\}, \tag{2.16}$$

wobei

$$(X-\eta)^+ := \begin{cases} (X-\eta), & \text{falls } (X-\eta) \geq 0 \\ 0 & \text{sonst} \end{cases}$$

Der α-Conditional Value-at-Risk reflektiert die "zu erwartenden Kosten in den (1-α) $\cdot$ 100% schlechtesten Fällen ". In [94] wurde gezeigt, dass (2.16) ein konvexes Minimierungsproblem in η mit der Lösungsmenge $[\eta_\alpha, \eta_\alpha^+]$ ist, wobei η_α^+ definiert ist als

$$\eta_\alpha^+ := \inf_{\eta\in\mathbb{R}}\{\eta : \mathbb{P}(X > \eta) < 1-\alpha\}.$$

Offenbar sind η_α und η_α^+ identisch, falls das vorgegebene α nicht auf ein konstantes Segment der Verteilungsfunktion trifft, vergleiche α_3 in Abbildung 2.4. In dem Fall ist die Lösungsmenge von (2.16) eindeutig. Trifft das α weder auf ein konstantes Segment, noch auf einen Zwischenraum - vergleiche α_1 in Abbildung 2.4 -, so ist der α-Conditional Value-at-Risk identisch mit dem bedingten Erwartungswert

$$\mathbb{E}[X|X \geq \mathcal{R}_{VaR_\alpha}(X)],$$

was insbesondere bei stetigen Verteilungen der Fall ist. Eine genauere Untersuchung des α-Conditional Value-at-Risk ist in [93, 94] zu finden.

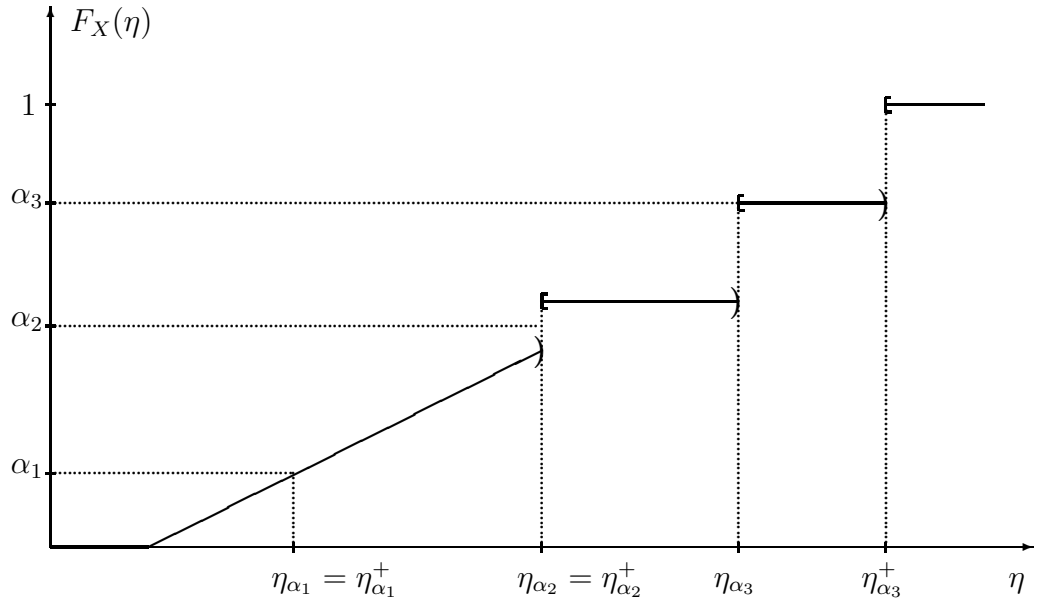


Abb. 2.4: Mögliche Varianten bei der Wahl des α's

Beispiel 2.6 (Excess Probability)
Die Excess Probability gibt die Wahrscheinlichkeit an, dass die Werte der Zufallsvariablen größer sind als ein vorgegebenes $\varphi \in \mathbb{R}$

$$\mathcal{R}_{EP_\varphi}(X) := \mathbb{P}(X > \varphi). \tag{2.17}$$

Beispiel 2.7 (Semideviation)
Die Semideviation gibt den Erwartungswert der Differenz zwischen allen Realisierungen von X, die größer als der Erwartungswert von X sind und dem Erwartungswert von X an.

$$\mathcal{R}_{SD}(X) := \mathbb{E}[(X - \mathbb{E}[X])^+]. \tag{2.18}$$

Beispiel 2.8 (Expected Excess)
Der Expected Excess gibt den Erwartungswert der Differenz zwischen allen Realisierungen von X, die größer als ein vorgegebenes $\varphi \in \mathbb{R}$ sind, und φ an.

$$\mathcal{R}_{EE_\varphi}(X) := \mathbb{E}[(X - \varphi)^+]. \tag{2.19}$$

Wie im vorangegangenen Kapitel erläutert, werden wir nun kurz auf den konzeptionellen Aspekt der Risikomodellierung eingehen.

Die Diskussion zur Charakterisierung von Risikomaßen wurde insbesondere durch Artzner et al. [8] angeregt. Die Autoren schlagen ein Axiomssystem für Risikomaße vor, welches den Bedürfnissen der Finanzwirtschaft Rechnung trägt. Risikomaße, die diese Axiome erfüllen, heißen kohärent. Der Conditional Value-at-Risk ist kohärent, siehe [1, 8, 89, 93]. Der Value-at-Risk verletzt hingegen die Bedingungen, siehe [8, 89], was zu einem gesteigerten Interesse des Conditional Value-at-Risk, bei der Minimierung finanzieller Risiken führte. Der Expected Excess und die Excess Probability sind nicht kohärent im Sinne von [8], was an der fest vorgegebenen Schranke φ liegt. In [71] wurde diesbezüglich eine Modifikation der Axiome vorgenommen, welche zum Begriff kohärent mit fester Schranke führte. Sowohl der Expected Excess als auch die Excess Probability erfüllen diese Bedingungen. Die Varianz und Semideviation sind keine kohärenten Risikomaße. In [38] wurde ebenfalls ein Axiomssystem vorgeschlagen, dessen geforderte Bedingungen schwächer sind als in [8] und konvexe Risikomaße definiert. Alle kohärenten Risikomaße sind auch konvexe Risikomaße.

Stochastische Dominanz [36, 68, 76, 79] ist eines der fundamentalen Konzepte der Entscheidungstheorie, bei dem Halbordnungen in $\mathcal{Z}$ definiert werden. Diese Halbordungen dienen als Entscheidungsgrundlage nach der Suche einer 'besten' Zufallsvariablen innerhalb einer gegebenen Familie und sind für $X, Y \in \mathcal{Z}$ folgendermaßen definiert:

$$X \leq_{FSD} Y \quad :\Longleftrightarrow \quad \forall x \in \mathbb{R}: \quad F_X(x) \geq F_Y(x), \tag{2.20}$$

$$X \leq_{SSD} Y \quad :\Longleftrightarrow \quad \forall x \in \mathbb{R}: \quad \mathbb{E}[(X-x)^+] \leq \mathbb{E}[(Y-x)^+], \tag{2.21}$$

wobei (2.20) als stochastische Ordnung ersten Grades, (2.21) als stochastische Ordnung zweiten Grades bezeichnet wird. Weitere wünschenswerte Eigenschaften speziell für Mean-Risk Modelle wurden von W. Ogryczak und A. Ruszczyński in [84, 85, 86] mithilfe dieser Halbordnungen entwickelt. Ein Mean-Risk Modell mit einem $\beta > 0$ heißt konsistent mit stochastischer Ordnung ersten bzw. zweiten Grades, falls aus $X \leq_{FSD} Y$ bzw. $X \leq_{SSD} Y$ folgt: $\mathbb{E}[X] + \beta\,\mathcal{R}[X] \leq \mathbb{E}[Y] + \beta\,\mathcal{R}[Y]$. In [86] wurde gezeigt, dass die Mean-Risk Modelle mit Excess Probability und Value-at-Risk konsistent mit stochastischer Ordnung ersten Grades für alle $\beta > 0$ und die Mean-

Risk Modelle mit Expected Excess und Conditional Value-at-Risk konsistent mit stochastischer Ordnung zweiten Grades für alle $\beta > 0$ sind. Die Semideviation ist konsistent mit stochastischer Ordnung ersten und zweiten Grades für $\beta = 1$, siehe [85]. Die Varianz ist im Allgemeinen nicht konsistent mit stochastischer Ordnung ersten oder zweiten Grades.

2.3.2 Polyedrische Risikomaße

Ein mächtiges Werkzeug bei der algorithmischen Behandlung mehrstufiger Optimierungsprobleme sind Dekompositionsverfahren, siehe [13, 54, 98, 99]. Wir wollen in dieser Arbeit die Szenariodekomposition nutzen, welche voraussetzt, dass das Wait-and See Problem, vergleiche Abbildung 2.2 in Abschnitt 2.2, separabel in $\omega \in \Omega$ ist. Diese Eigenschaft ist insbesondere beim erwartungswertbasierten Modell gegeben.

Ein anderes Verfahren ist die geographische- oder komponentenweise Dekomposition, bei der in Abhängigkeit eines konkreten Anwendungsproblems schwach koppelnde Restriktionen relaxiert werden, so dass das Problem in Teilprobleme zerfällt. Die relaxierten Bedingungen können dann effizient mithilfe des Lagrange'schen Dualproblems wiederhergestellt werden, siehe etwa [29, 83].

Wir werden in diesem Kapitel untersuchen, inwieweit die Separabilität in $\omega \in \Omega$ durch die Hinzunahme von Risikomaßen erhalten bleibt. Dazu betrachten wir zunächst den zweistufigen Fall ($T = 2$). Eine Klassifizierung von Risikomaßen die in [34] vorgestellt wurde ist in diesem Zusammenhang sehr hilfreich.

Definition 2.9 (Ein-periodisches polyedrisches Risikomaß)
Ein Risikomaß $\mathcal{R}$ definiert auf $L_p(\Omega, \mathcal{F}, \mathbb{P})$, mit $p \in [1, \infty]$ heißt polyedrisch, falls $k_1, k_2 \in \mathbb{N}$, $c_1, g_1 \in \mathbb{R}^{k_1}$, $c_2, g_2 \in \mathbb{R}^{k_2}$, ein nichtleeres Polyeder $Y_1 \subseteq \mathbb{R}^{k_1}$ und ein polyedrischer Kegel $Y_2 \subseteq \mathbb{R}^{k_2}$ so existieren, dass für alle $z \in L_p(\Omega, \mathcal{F}, \mathbb{P})$ gilt:

$$\mathcal{R}(z) = \inf \left\{ d_1^\top y_1 + \mathbb{E}[c_2^\top y_2] \;\middle|\; \begin{array}{l} y_1 \in Y_1, \\ y_2 \in L_p(\Omega, \mathcal{F}, \mathbb{P}),\ y_2 \in Y_2, \\ g_1^\top y_1 + g_2^\top y_2 = -z \end{array} \right\}. \tag{2.22}$$

Obige Definition weicht leicht von der ursprünglichen Definition in [34] ab, da hier z durch $-z$ als rechte Seite eingesetzt wurde. Anders als in [34] werden hier kleinere

Werte bevorzugt.
Ein polyedrisches Risikomaß kann als ein spezielles zweistufiges Erwartungswertproblem, mit zufälligen rechten Seiten dargestellt werden. Der Begriff "polyedrisch" ist motiviert durch die Tatsache, dass für endliches Ω das Risikomaß $\mathcal{R}$ eine polyedrische Funktion auf dem $\mathbb{R}^{\#\Omega}$ ist.

Bemerkung 2.10 *Die gewichtete Summe aus Erwartungswert $\mathbb{E}$ und einem polyedrischen Risikomaß $\mathcal{R}$ ist ebenfalls polyedrisch. Für ein $\beta \in [0,1]$ erfüllt die gewichtete Summe $\hat{\mathcal{R}} := \beta\,\mathcal{R} + (1-\beta)\mathbb{E}$ die entsprechenden Bedingungen, falls $\hat{d}_1 := \beta\, d_1 - (1-\beta)\, g_1$, $\hat{d}_2 := \beta\, d_2 - (1-\beta)\, g_2$, $\hat{g}_1 := g_1$ und $\hat{g}_2 := g_2$ gesetzt wird. Vergleiche auch [34]. Damit brauchen Mean-Risk Modelle nicht gesondert betrachtet zu werden.*

Hier zwei Beispiele für polyedrische Risikomaße:

Beispiel 2.11 *Der Conditional Value-at-Risk $\mathcal{R}_{CVaR_\alpha}$ erfüllt 2.22, denn es gilt:*

$$\mathcal{R}_{CVaR_\alpha}(z) = \inf_{\eta\in\mathbb{R}} \left\{ \eta + \frac{1}{1-\alpha}\mathbb{E}\left[(z-\eta)^+\right] \right\}$$

$$= \inf_{y_1\in\mathbb{R}, y_2\in\mathbb{R}^2} \left\{ y_1 + \mathbb{E}\left[\frac{1}{1-\alpha}\, y_2^{(1)}\right] \;\middle|\; \begin{array}{l} -y_1 - y_2^{(1)} + y_2^{(2)} = -z, \\ \\ y_2^{(1)} \geq 0,\ y_2^{(2)} \geq 0,\ (y_2^{(1)}, y_2^{(2)}) \in L_p(\Omega, \mathcal{F}, \mathbb{P}) \end{array} \right\}$$

$$= \inf \left\{ d_1^\top y_1 + \mathbb{E}\left[d_2^\top y_2\right] \;\middle|\; \begin{array}{l} g_1^\top y_1 + g_2^\top y_2 = -z, \\ \\ y_1 \in Y_1,\ y_2 \in Y_2,\ y_2 \in L_p(\Omega, \mathcal{F}, \mathbb{P}) \end{array} \right\}$$

wobei in der letzten Gleichung $d_1 := 1$, $d_2 := (\frac{1}{1-\alpha}, 0)$, $g_1 := -1$, $g_2 := (-1, 1)$, $Y_1 := \mathbb{R}$, $Y_2 := \mathbb{R}^2_+$ gesetzt wurde.

Beispiel 2.12 *Das Expected Excess ist ein polyedrisches Risikomaß, denn es gilt:*

$$\mathcal{R}_{EE_\varphi}(z) = \left\{ \mathbb{E}\left[(z-\varphi)^+\right] \right\}$$

$$= \inf_{y_1\in\mathbb{R}, y_2\in\mathbb{R}^2} \left\{ y_1 + \mathbb{E}\left[y_2^{(1)}\right] \;\middle|\; \begin{array}{l} -y_1 - y_2^{(1)} + y_2^{(2)} = -z, \\ \\ y_2^{(1)} \geq 0,\ y_2^{(2)} \geq 0,\ (y_2^{(1)}, y_2^{(2)}) \in L_p(\Omega, \mathcal{F}, \mathbb{P}) \end{array} \right\}$$

$$= \inf \left\{ d_1^\top y_1 + \mathbb{E}\left[d_2^\top y_2\right] \;\middle|\; \begin{array}{l} g_1^\top y_1 + g_2^\top y_2 = -z, \\ \\ y_1 \in Y_1,\ y_2 \in Y_2,\ y_2 \in L_p(\Omega, \mathcal{F}, \mathbb{P}) \end{array} \right\}$$

wobei in der letzten Gleichung $d_1 := 0$, $d_2 := (1,0)$, $g_1 := -1$, $g_2 := (-1,1)$, $Y_1 := \varphi$, $Y_2 := \mathbb{R}^2_+$ *gesetzt wurde.*

Bis jetzt wurden Risikomaße für eine Zufallsvariable z betrachtet. Im mehrstufigen Modell betrachtet man einen stochastischen Kostenprozess, vergleiche (2.5) oder (2.7). Wie bereits am Ende des Abschnitts 2.2 diskutiert, ist es vom Entscheidungsträger abhängig, ob die Kosten in allen Stufen also $(z_1, \ldots, z_T)$ oder nur die finalen Kosten z_T von Interesse sind. Im letzten Fall bleibt es bei der Betrachtung einer einzigen Zufallsvariablen. Wir werden in dieser Arbeit Modelle entwickeln, in denen das Risiko in allen Stufen berücksichtigt wird.

Definition 2.13 (Mehrperiodisches polyedrisches Risikomaß)
Ein Risikomaß $\mathcal{R}$ *definiert auf* $\times_{t=1}^T L_p(\Omega, \mathcal{F}_t, \mathbb{P})$*, mit* $p \in [1, \infty]$ *heißt mehrperiodisch polyedrisch, falls* $k_t \in \mathbb{N}$, $d_t \in \mathbb{R}^{k_t}$, $t = 1, \ldots, T$, $g_{t,\tau} \in \mathbb{R}^{k_{t-\tau}}$, $t = 1, \ldots, T$, $\tau = 0, \ldots, t-1$ *ein nichtleeres Polyeder* $X_1 \subseteq \mathbb{R}^{k_1}$ *und polyedrische Kegel* $X_t \subseteq \mathbb{R}^{k_t}$, $t = 2, \ldots, T$ *so existieren, dass für alle* $z := (z_1, \ldots, z_T) \in \times_{t=1}^T L_p(\Omega, \mathcal{F}_t, \mathbb{P})$ *gilt:*

$$\mathcal{R}(Z) = \inf \left\{ \mathbb{E}[\sum_{t=1}^T d_t^\top x_t] \;\middle|\; \begin{array}{l} x_t \in L_p(\Omega, \mathcal{F}_t, \mathbb{P}; \mathbb{R}^{k_t}) \\ x_t \in X_t, \\ \sum_{\tau=0}^{t-1} g_{t,\tau}^\top x_{t-\tau} = -z_t,\ t = 1, \ldots, T \end{array} \right\}. \tag{2.23}$$

In [34] werden mehrere Spezifikationen von (2.23) diskutiert, welche zu mehrstufigen Risikomaßen verschiedener Komplexität führen. Insbesondere werden Beispiele vorgeführt, in denen der Informationsfluss, dargestellt durch die Filtration $\{\mathcal{F}_t\}_{t=1}^T$, mit berücksichtigt wird. Wir werden in dieser Arbeit die in Abschnitt 2.3.1 vorgestellten Risikomaße durch gewichtete Summen zu mehrperiodischen Risikomaßen

erweitern. Unter anderem führt die Erweiterung des Conditional Value-at-Risk zu einem mehrperiodischen polyedrischen Risikomaß.

Beispiel 2.14 (Mehrperiodischer Conditional Value-at-Risk, [34])
Es sei $z := (z_1, \dots, z_T) \in \times_{t=1}^{T} L_p(\Omega, \mathcal{F}_t, \mathbb{P})$. *Mit nicht-negativen Gewichten* $\gamma_1, \dots, \gamma_T$ *und vorgegebenen Wahrscheinlichkeiten* $\alpha := (\alpha_2, \dots, \alpha_T) \in (0,1)^{T-1}$ *summieren wir den Conditional Value-at-Risk in den verschiedenen Zeitperioden*

$$\mathcal{R}_{MCVaR_\alpha}(z) := \sum_{t=2}^{T} \gamma_t \, \mathcal{R}_{CVaR_{\alpha_t}}(z_t). \tag{2.24}$$

Dann gilt

$$\mathcal{R}_{MCVaR_\alpha}(z) = \sum_{t=2}^{T} \gamma_t \inf_{\eta_t \in \mathbb{R}} \{\eta_t + \frac{1}{1-\alpha_t} \mathbb{E}[(z_t - \eta_t)^+]\}$$

$$= \inf_{(\eta_2, \dots, \eta_T) \in \mathbb{R}^{T-1}} \left\{ \sum_{t=2}^{T} \gamma_t \left(\eta_t + \frac{1}{1-\alpha_t} \mathbb{E}[(z_t - \eta_t)^+] \right) \right\}$$

$$= \inf \left\{ \sum_{t=2}^{T} \gamma_t \left(\eta_t + \frac{1}{1-\alpha_t} \mathbb{E}[y_t^{(1)}] \right) \;\middle|\; \begin{array}{l} -\eta_t - y_t^{(1)} + y_t^{(2)} = -z_t, \\ y_t^{(1)} \geq 0, \; y_t^{(2)} \geq 0, \; H_t(x_t) = 0, \\ t = 2 \dots, T, \; (\eta_2, \dots, \eta_T) \in \mathbb{R}^{T-1} \end{array} \right\}$$

$$= \inf \left\{ \mathbb{E}\left[\sum_{t=1}^{T} d_t^\top y_t \right] \;\middle|\; y_t \in Y_t, \; H_t(y_t) = 0, \; \sum_{\tau=0}^{t-1} e_{t,\tau}^\top y_{t-\tau} = -z_t, \; t = 1 \dots, T \right\}$$

mit $k_1 = T$, $k_t = 2$, $t = 2, \dots, T$, $d_1 = (0, \gamma_2, \dots, \gamma_T)$, $d_t = (\frac{\gamma_t}{1-\alpha_t}, 0)$, $t = 2, \dots, T$, $g_{1,0} = e_1$, $g_{t,0} = (-1, 1)$, $t = 2, \dots, T$, $g_{t,t-1} = e_t$, $t = 2, \dots, T$, $g_{t,\tau} = 0$, $\tau = 1, \dots, t-2$, $t = 3, \dots, T$, $Y_1 = R^T$, $Y_t = \mathbb{R}^2_+$, $t = 2, \dots, T$, *wobei* e_t *den* $t-$*ten Einheitsvektor im* $\mathbb{R}^T$ *repräsentiert.*

Auch der Expected Excess kann durch eine gewichtete Summe zu einem mehrperiodischen, polyedrischen Risikomaß erweitert werden.

Beispiel 2.15 (Mehrperiodischer Expected Excess)
Es sei $z := (z_1, \ldots, z_T) \in \times_{t=1}^{T} L_p(\Omega, \mathcal{F}_t, \mathbb{P})$. *Mit nicht-negativen Gewichten* $\gamma_1, \ldots, \gamma_T$ *und vorgegebenen* $\varphi := (\varphi_2, \ldots, \varphi_T) \in \mathbb{R}^{T-1}$ *summieren wir das Expected Excess in den verschiedenen Zeitperioden*

$$\mathcal{R}_{MEE_\varphi}(z) := \sum_{t=2}^{T} \gamma_t \mathcal{R}_{EE_{\varphi t}}(z_t). \tag{2.25}$$

Dann gilt

$$\mathcal{R}_{EE}(z) = \sum_{t=2}^{T} \gamma_t \, \mathbb{E}[(z_t - \varphi_t)^+]$$

$$= \inf \left\{ \sum_{t=2}^{T} \gamma_t \left(\mathbb{E}[y_t^{(1)}] \right) \;\middle|\; \begin{array}{l} -\varphi_t - y_t^{(1)} + y_t^{(2)} = -z_t, \\ x_t^{(1)} \geq 0, \; y_t^{(2)} \geq 0, \; H_t(y_t) = 0, \\ t = 2, \ldots, T \end{array} \right\}$$

$$= \inf \left\{ \mathbb{E}\left[\sum_{t=1}^{T} d_t^\top y_t \right] \;\middle|\; y_t \in Y_t, \; H_t(y_t) = 0, \; \sum_{\tau=0}^{t-1} g_{t,\tau}^\top y_{t-\tau} = -z_t, \; t = 1 \ldots, T \right\}$$

mit $k_1 = T$, $k_t = 2$, $t = 2, \ldots, T$, $d_1 = (0, \ldots, 0)$, $d_t = \gamma_t$, $t = 2, \ldots, T$, $g_{1,0} = e_1$, $g_{t,0} = (-1, 1)$, $t = 2, \ldots, T$, $g_{t,t-1} = e_t$, $t = 2, \ldots, T$, $g_{t,\tau} = 0$, $\tau = 1, \ldots, t-2$, $t = 3, \ldots, T$, $Y_1 = (0, \eta_2, \ldots, \eta_T)$, $Y_t = \mathbb{R}^2_+$, $t = 2, \ldots, T$, *wobei* e_t *den* $t-ten$ *Einheitsvektor im* $\mathbb{R}^T$ *repräsentiert.*

Wie in Kapitel 3.2 sei z_t die akkumulierte Summenvariable, vergleiche (2.5). Somit kann analog zu (2.6) das Kompensationsproblem mit einem gegebenen Risikomaß $\mathcal{R}$ dargestellt werden durch

$$\min \{ \mathcal{R}(z_1, \ldots, z_T) \, | x \in \mathcal{X}(\xi) \} . \tag{2.26}$$

Ist $\mathcal{R}$ ein polyedrisches Risikomaß, so kann es als ein spezielles Erwartungswertproblem dargestellt werden. Benutzt man diese Darstellung von $\mathcal{R}$, so ergibt sich

folgendes Optimierungsproblem

$$\min \left\{ \mathbb{E}\left[\sum_{t=1}^{T} d_t^\top y_t\right] \;\middle|\; \begin{array}{l} x \in \mathcal{X}(\xi), \\ y_t \in Y_t,\ H_t(y_t) = 0, \\ \sum_{\tau=0}^{t-1} g_{t,\tau}^\top y_{t-\tau} = -z_t,\ t = 1, \ldots, T \end{array} \right\} \tag{2.27}$$

In [34] wurde die folgende Äquivalenzaussage zwischen (2.26) und (2.27) bewiesen.

Proposition 2.16 ([34], Proposition 4.1)
Das Problem (2.26) bezüglich x- und das Problem (2.27) bezüglich des Paares (x, y) zu minimieren, ist im folgendem Sinne äquivalent:
i) Die optimalen Werte beider Probleme stimmen überein.
ii)Das Paar (x^*, y^*) ist eine Lösung von (2.27) genau dann wenn x^* das Problem (2.26) löst und y^* eine Lösung des Minimierungsproblems $\mathcal{R}\left(\left(\sum_{\tau=1}^{t} c_\tau(\xi_\tau)^\top x_\tau^*\right)_{t=1}^{T}\right)$ ist.

Folgende wichtige Aussage kann im Hinblick auf die Szenariodekomposition bei polyedrischen Risikomaßen gemacht werden:

Lemma 2.17 *Die Relaxation der Nichtantizipativitätsbedingungen in (2.27) führt zu einem Wait-and-See Problem, welches separabel in $\omega \in \Omega$ ist.*

Beweis
Ohne Nichtantizipativitätsbedingungen gibt es keine Nebenbedingungen in (2.27) die Komponenten von $\xi(\cdot)$ mit unterschiedlichen $\omega, \omega' \in \Omega$ enthalten. Dieses erlaubt das Vertauschen von Integrand und Minimum,

$$\min \left\{ \mathbb{E}\left[\sum_{t=1}^{T} d_t(\xi_t)^\top y_t\right] \;\middle|\; \begin{array}{l} x_t \in X_t,\ \ y_t \in Y_t \\ \sum_{\tau=1}^{t} A_{t\tau}(\xi_t) x_\tau \le b_t(\xi_t), \\ \sum_{\tau=0}^{t-1} \omega_{t,\tau}^\top y_{t-\tau} = -z_t,\ t = 1 \ldots, T \end{array} \right\}$$

$$= \mathbb{E}\left[\min \left\{ \sum_{t=1}^{T} d_t(\xi_t)^\top y_t \;\middle|\; \begin{array}{l} x_t \in X_t,\ \ y_t \in Y_t \\ \sum_{\tau=1}^{t} A_{t\tau}(\xi_t) x_\tau \le b_t(\xi_t), \\ \sum_{\tau=0}^{t-1} \omega_{t,\tau}^\top y_{t-\tau} = -z_t,\ t = 1 \ldots, T \end{array} \right\} \right]$$

siehe beispielsweise Theorem 14.60 in [95]. Das Minimum kann somit für jedes $\omega \in \Omega$ einzeln berechnet werden. □

Obwohl es für die Excess Probability keine Darstellung als polyedrisches Risikomaß gibt, kann gezeigt werden, das auch dieses Risikomaß die gewünschte Separabilität aufweist.

Beispiel 2.18 (Mehrperiodische Excess Probability)
Wie in den vorangegangenen Beispielen summieren wir das Excess Probability in den verschiedenen Zeitperioden mit positiven Gewichten $\gamma_2, \ldots, \gamma_T$ auf. Mit vorgegebenen Schranken $\varphi := (\varphi_2, \ldots, \varphi_T) \in \mathbb{R}^{T-1}$ definieren wir eine mehrperiodische Erweiterung des Excess Probability durch

$$\mathcal{R}_{MEP_\varphi}(z) := \sum_{t=2}^{T} \gamma_t \, \mathcal{R}_{EP_{\varphi_t}}(z_t)$$

und betrachten das Optimierungsproblem

$$\min \left\{ \mathcal{R}_{MEP}(z) | x \in \mathcal{X}(\xi) \right\}.$$

Ist die Menge $\{Z_t | x \in \mathcal{X}(\xi)\}$, $t = 2, \ldots, T$, $\mathbb{P} - f.s.$ beschränkt, existiert eine obere Schranke M>0, so dass obiges Problem dargestellt werden kann durch

$$\min \left\{ \sum_{t=2}^{T} \gamma_t \, \mathbb{P}(z_t > \varphi_t) | x \in \mathcal{X}(\xi) \right\}$$

$$= \min \left\{ \sum_{t=2}^{T} \gamma_t \, \mathbb{E}\left[\mathbf{1}_{(z_t > \varphi_t)} \right] | x \in \mathcal{X}(\xi) \right\}$$

$$= \min \left\{ \sum_{t=2}^{T} \gamma_t \, \mathbb{E}\left[u_t \right] \;\middle|\; \begin{array}{l} x \in \mathcal{X}(\xi), \\ z_t - \varphi_t \leq M \, u_t, \;\; u_t \in \{0,1\}, \\ H(u_t) = 0, \;\; t = 2, \ldots, T \end{array} \right\}. \tag{2.28}$$

Hier ist die Indikatorfunktion wie folgt definiert:

$$\mathbf{1}_{(z_t > \varphi_t)} := \begin{cases} 1, & \text{falls } \; z_t > \varphi_t \\ 0 & \text{sonst} \end{cases}.$$

Relaxiert man die Nichtantizipativitätsbedingungen in (2.28), zerfällt das Problem über Ω.

Nun einige Beispiele von Risikomaßen, die nicht über Ω separabel sind

Beispiel 2.19 (Mehrperiodischer Value-at-Risk)
Wir betrachten die mehrstufige Erweiterung

$$\mathcal{R}_{MVaR_\alpha}(z) := \sum_{t=2}^{T} \gamma_t \, \mathcal{R}_{VAR_{\alpha_t}}(z_t)$$

mit positiven Gewichten $\gamma_2, \ldots, \gamma_t$ und vorgegebenen Wahrscheinlichkeiten $\alpha := (\alpha_2, \ldots, \alpha_T) \in (0,1)^{T-1}$. Wie im vorigen Beispiel nehmen wir an, dass $\{z_t | x \in \mathcal{X}(\xi)\}$, $t = 2, \ldots, T$, $\mathbb{P} - f.s.$ beschränkt ist mit einer oberen Schranke M>0. Dann definieren wir das Optimierungsproblem

$$\min \left\{ \mathcal{R}_{MVaR_\alpha}(z) | x \in \mathcal{X}(\xi) \right\}.$$

$$= \min \left\{ \sum_{t=2}^{T} \gamma_t \, \eta_t \;\middle|\; \begin{array}{l} x \in \mathcal{X}(\xi), \\ \mathbb{P}(z_t > \eta_t) \leq 1 - \alpha_t, \\ \eta_t \in \mathbb{R}, \;\; H(\eta_t) = 0, \;\; t = 2, \ldots, T \end{array} \right\}.$$

Durch die Restriktionen $\mathbb{P}(z_t > \eta_t) \leq 1 - \alpha_t$, $t = 2, \ldots, T$ werden mehrere ω miteinander gekoppelt. Damit ist das Wait-and-See Problem nicht separabel.

Beispiel 2.20 (Mehrperiodische Semideviation)
Wir betrachten die mehrstufige Erweiterung

$$\mathcal{R}_{MSD}(z) := \sum_{t=2}^{T} \gamma_t \, \mathcal{R}_{SD}(z_t)$$

mit positiven Gewichten $\gamma_2, \ldots, \gamma_t$. Es gilt

$$\min \left\{ \mathcal{R}_{MSD}(z) | x \in \mathcal{X}(\xi) \right\}$$

$$= \min \left\{ \sum_{t=2}^{T} \gamma_t \, \mathbb{E}[\eta_t] \;\middle|\; \begin{array}{l} x \in \mathcal{X}(\xi), \\ z_t - \mathbb{E}[z_t] \leq \eta_t, \;\; \eta \geq 0, \\ \eta_t \in \mathbb{R}, \;\; H(\eta_t) = 0, \;\; t = 2, \ldots, T \end{array} \right\}.$$

Hier koppelt der Erwartungswert in den Restriktionen $z_t - \mathbb{E}[z_t] \leq \eta_t$, $t = 2, \ldots, T$ verschiedene ω miteinander

Im letzten Beispiel wird nicht nur die Separabilität des Wait-and-See Problems in Ω verletzt, sondern auch die Linearität, die in allen vorherigen Beispielen erhalten blieb.

Beispiel 2.21 (Mehrperiodische Varianz)
Mit positiven Gewichten $\gamma_2, \ldots, \gamma_t$ betrachten wir die mehrstufige Erweiterung der Varianz durch

$$\mathcal{R}_{MVar}(z) := \sum_{t=2}^{T} \gamma_t \, \mathcal{R}_{VAR}(z_t).$$

Es ist

$$\min \{\mathcal{R}_{MVar}(z) | x \in \mathcal{X}(\xi)\}$$
$$= \min \left\{ \sum_{t=2}^{T} \gamma_t \, \mathbb{E}[\eta_t] \;\middle|\; \begin{array}{l} x \in \mathcal{X}(\xi), \\ (z_t - \mathbb{E}[z_t])^2 = \eta_t, \\ \eta_t \in \mathbb{R}, \;\; H(\eta_t) = 0, \;\; t = 2, \ldots, T \end{array} \right\}.$$

Durch die Restriktionen $(z_t - \mathbb{E}[z_t])^2 = \eta_t$, $t = 2, \ldots, T$ werden mehrere ω miteinander gekoppelt und Nichtlinearitäten müssen zusätzlich berücksichtigt werden.

Die folgende Tabelle fasst noch einmal die behandelten Eigenschaften der hier vorgestellten Risikomaße zusammen, wobei die mehrstufige Erweiterung durch die gewichtete Summe definiert wurde. Der Eintrag "–" bedeutet, dass noch keine Resultate bekannt sind.

Mean-Risk Modell	Linearität bleibt erhalten	Separabel in Ω	Polyedrisches Risikomaß
R_{MVar}	NEIN	NEIN	NEIN
R_{MVaR_α}	JA	NEIN	NEIN
$\mathrm{R}_{MCVaR_\alpha}$	JA	JA	JA
R_{MEP_φ}	JA	JA	–
R_{MSD}	JA	NEIN	NEIN
R_{MEE_φ}	JA	JA	JA

Tabelle 2.5: Polyedrische und Separabele Risikomaße

Kapitel 3

Algorithmen

3.1 Allgemeines

Inwieweit die obigen Probleme numerisch berechenbar sind, hängt entscheidend von dem zugrunde liegenden Wahrscheinlichkeitsmaß ab. Man denke beispielsweise an stetige Verteilungen, bei denen die exakte Berechnung aufgrund der enthaltenden Integrale nur sehr schwierig bis unmöglich ist. Approximation durch diskrete Wahrscheinlichkeitsmaße ist in diesem Zusammenhang ein bewährtes Mittel, um solche Probleme numerisch behandeln zu können. Dann stellt sich die Frage, welchen Einfluss diese Approximation auf die Güte der optimalen Lösung hat. Diese Frage motiviert Stabilitätsuntersuchungen. Man möchte sicherstellen, dass "kleine" Störungen bei der Approximation des Risikomaßes nur "kleine " Störungen bei der gefundenen Lösung verursachen.
Die Notation Stabilitätsuntersuchung in der stochastischen Optimierung wurde erstmals in [14, 62] benutzt. Als einleitende Lektüre sei auf Kapitel 2 in [96] verwiesen. Resultate findet man unter anderem für zweistufige erwartungswertbasierte Probleme in [95, 96, 105, 106, 107, 113]; für zweistufige risikobasierte Probleme in [109, 110]. Mehrstufige erwartungswertbasierte Probleme wurden in [37, 52, 118] untersucht und mehrstufige risikobasierte Probleme in [34].
Eine praktische Anwendung dieser Stabilitätsuntersuchungen betrifft die Szenarioreduktion im Fall endlich vieler Szenarien, siehe beispielsweise [51, 53, 78]. Da es in diesem Fall häufig zu Problemen kommt, die aufgrund ihrer Größe nicht mehr

zu berechnen sind, sucht man durch die Verringerung der Szenarienanzahl ebenfalls einen Kompromiß zwischen Berechenbarkeit der Probleme und Genauigkeit der berechneten Lösung.
Es sei im Weiteren vorausgesetzt, dass die Wahrscheinlichkeitsverteilung nur endlich viele Szenarien besitzt. In diesem Fall gehen die angesprochenen Integrale in endliche Summen über und man erhält ein endlichdimensionales gemischt-ganzzahliges Optimierungsproblem. Wie bereits angedeutet hängt die Größe dieser Probleme entscheidend von der Anzahl der Szenarien ab, so dass es für Standardsolver wie beispielsweise CPLEX [57] oder Xpress-MP [28] sehr schwierig ist, diese zu lösen. Es ist deshalb notwendig, die aus den Modellen entstehende Struktur zu nutzen, um problemspezifische Algorithmen zu entwerfen.
Viele Verfahren, die zum Lösen dieser Probleme genutzt werden, wie etwa Schnittebenenverfahren, Branch-and-Bound Verfahren, Dekomposition und Lagrange Relaxation, stammen aus der ganzzahligen Optimierung, siehe beispielsweise [81, 88, 120]. Da zweistufige Probleme länger als die allgemeineren mehrstufigen Probleme untersucht wurden, ist die Anzahl der entwickelten Algorithmen entsprechend größer. Es folgt nun eine Auswahl an bereits existierenden Algorithmen. Eine abstraktere Untersuchung algorithmischer Methoden der linearen- sowie linear gemischtganzahligen stochastischen Optimierung findet man unter anderem in [18, 54, 60, 69, 99, 111]. Zweistufige Probleme mit binären Erststufenvariablen werden in [67] behandelt. Ein Algorithmus für den allgemeinen gemischt-ganzzahligen zweistufigen Fall wird in [20] vorgestellt. Der Algorithmus basiert auf Szenariodekomposition, Lagrange-Relaxation und einem Branch-and-Bound Schema. Im mehrstufigen Fall sei auf [5, 6] für den rein binären Fall verwiesen. Hier wird ebenfalls die Szenariodekomposition genutzt und ein Branch-and-Cut Verfahren initialisiert. Ein Schnittebenenverfahren für allgemeine gemischt ganzzahlige Probleme wurde in [44] entwickelt. Ein Algorithmus der speziell für Investitionsprobleme mit binären Entscheidungs- und stetigen Variablen entwickelt wurde und auf einem Branch-and-Bound Verfahren basiert, wurde in [3] konzipiert. Eine Auswahl weiterer Algorithmen und viele praktische Anwendungen wurde kürzlich in [117] veröffentlicht.
Resultate über die Komplexität stochastischer Optimierungsprobleme findet man in [32, 82, 112].

In Kapitel 3.2 wird das mehrstufige Erwartungswertmodell (2.3) für den Fall einer endlichen, diskreten Wahrscheinlichkeitsverteilung vorgestellt. Wir werden sehen, dass wir ein gemischt-ganzzahliges Optimierungsproblem erhalten, dessen Nebenbedingungsmatrix eine Blockstruktur aufweist.

In Abschnitt 3.2.1 erweitern wir das Erwartungswertmodell zu dem in Kapitel 2.3.1 vorgestellten Mean-Risk Modell für den diskreten Fall. Es wird gezeigt, dass in den Optimierungsproblemen die Blockstruktur erhalten bleibt, falls insbesondere polyedrische Risikomaße benutzt werden.

In Abschnitt 3.3 wird auf das grundlegende Lösungsverfahren eingegangen, das den Algorithmen bb-tree und bb-hash als Basis dient. Die konkreten Algorithmen werden dann in Abschnitt 3.3.1 bzw. Abschnitt 3.3.2 vorgestellt und anhand eines Beispiels erläutert. In Abschnitt 3.3.3 werden die wesentlichen Unterschiede der beiden Algorithmen gegenübergestellt und diskutiert.

3.2 Diskreter Fall

Wir betrachten nun eine Ereignismenge mit $S \in \mathbb{N}$ Szenarien, d.h. $\Omega = \{\omega_1, \ldots, \omega_S\}$. Diese Szenarien haben die Wahrscheinlichkeiten $\pi_s := \mathbb{P}(\{\omega_s\})$, $s = 1, \ldots, S$. Des Weiteren werden folgende Notationen benutzt: $\xi^s := \xi_t(\omega_s)$, $c_t^s := c_t(\xi_t(\omega_s))$, $b_t^s := b_t(\xi_t(\omega_s))$, $x_t^s := x_t(\xi_t(\omega_s))$ und $A_{t\tau}^s := A_{t\tau}(\xi_t(\omega_s))$, $\tau = 1, \ldots, t$, $t = 1, \ldots, T$, $s = 1, \ldots, S$.

Die σ-Algebra $\mathcal{F}$ ist nun die Potenzmenge 2^Ω von Ω. Zu jeder Subalgebra $\mathcal{F}_t$, $t = 1, \ldots, T$ existiert eine endliche Familie $\mathcal{E}_t \subseteq 2^\Omega$, so dass $\mathcal{E}_t$ eine Partition von Ω ist und $\mathcal{F}_t$ die von $\mathcal{E}_t$ erzeugte σ-Algebra. Das heißt

$$\sigma(\mathcal{E}_t) = \mathcal{F}_t, \ t = 1, \ldots, T. \tag{3.1}$$

Da $\mathcal{F}_t \subseteq \mathcal{F}_{t+1}$, $t = 1, \ldots, T-1$ ist jedes Element von $\mathcal{E}_t$ die Vereinigung aus Elementen von $\mathcal{E}_{t+1}$. Die Anzahl der Elemente von $\mathcal{E}_t$ stimmt mit den bis zum Zeitpunkt t unterscheidbaren Ereignissen überein. Insbesondere gilt also wegen den Voraussetzungen $\mathcal{F}_1 = \{\emptyset, \Omega\}$ und $\mathcal{F}_T = \mathcal{F}$, dass $\mathcal{E}_1 = \{\Omega\}$ und $\mathcal{E}_T = \{\{\omega_1\}, \ldots, \{\omega_S\}\}$. Die Elemente von $\mathcal{E}_t$, $t = 1, \ldots, T$ können also mit den Knoten $\mathcal{N}$ eines Szenariobaumes identifiziert werden, siehe Abbildung 4.1. Hier ist ein Szenariobaum mit sechs

Szenarien und fünf Zeitstufen dargestellt. Ein Szenario ist gerade der Weg vom Wurzelknoten zu einem Endknoten.

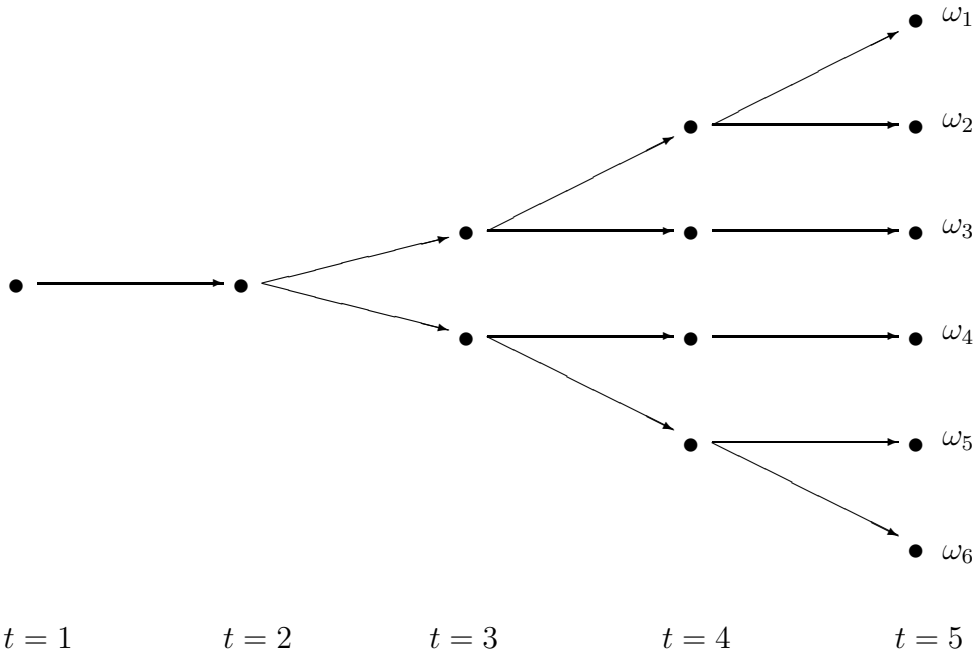


Abb. 4.1: Szenariobaum

Konkret hat man in diesem Beispiel $\mathcal{E}_1 = \{\Omega\}$, $\mathcal{E}_2 = \{\Omega\}$, $\mathcal{E}_3 = \{\{\omega_1, \omega_2, \omega_3\}, \{\omega_4, \omega_5, \omega_6\}\}$, $\mathcal{E}_4 = \{\{\omega_1, \omega_2\}, \{\omega_3\}, \{\omega_4\}, \{\omega_5, \omega_6\}\}$, $\mathcal{E}_5 = \{\{\omega_1\}, \{\omega_2\}, \{\omega_3\}, \{\omega_4\}, \{\omega_5\}, \{\omega_6\}\}$.

Die Nichtantizipativitätsbedingungen $H_t(x_t) = x_t - \mathbb{E}[x_t|\mathcal{F}_t] = 0$ in (2.3) stellen sich im diskreten Fall wie folgt dar: Für den bedingten Erwartungswert bezüglich der σ-Algebra $\mathcal{F}_t$, $t = 1, \ldots, T$ gilt ([97]):

$$\begin{aligned}\mathbb{E}[x_t|\mathcal{F}_t] &= \sum_{E\in\mathcal{E}_t} \frac{1}{\mathbb{P}(E)} \int_E x_t(\omega)\mathbb{P}(d\omega)\mathbf{1}_E \\ &= \sum_{E\in\mathcal{E}_t} \left(\sum_{\substack{s=1 \\ \omega_s\in E}}^{S} \pi_s \right)^{-1} \left(\sum_{\substack{s=1 \\ \omega_s\in E}}^{S} \pi_s x_t^s \right) \mathbf{1}_E \end{aligned} \tag{3.2}$$

Damit sind die Nichantizipativitätsbedingungen im diskreten Fall äquivalent zum

linearen Gleichungssystem:

$$x_t^\delta = \sum_{\substack{E\in\mathcal{E}_t \\ \omega_\delta\in E}} \left(\sum_{\substack{s=1 \\ \omega_s\in E}}^{S} \pi_s \right)^{-1} \left(\sum_{\substack{s=1 \\ \omega_s\in E}}^{S} \pi_s x_t^s \right), \quad \delta = 1,\dots,S, \;\; t = 1,\dots,T, \tag{3.3}$$

hinter dem sich die die folgenden Identitäten verbergen:

$$\forall t \in \{1,\dots,T\} : x_t^\delta = x_t^s \;\; \forall \delta, s \in \{1,\dots,S\} \text{ mit } \exists E \in \mathcal{E}_t : \omega_\delta, \omega_s \in E \tag{3.4}$$

Aus dem Entscheidungskontext bedeuten diese Gleichheitsbedingungen, dass zwei Entscheidungen x_t^s und $x_t^{s'}$ zu einem festen Zeitpunkt t für die Szenarien $s, s' \in \{1,\dots,S\}$ identisch sein müssen, solange diese nicht unterscheidbar sind. Zu welchem Zeitpunkt verschiedene Szenarien unterscheidbar sind, wird durch die σ-Algebren $\mathcal{F}_t$, $t = 1,\dots,T$ festgelegt. In Abbildung 4.2 wird diese Situation graphisch für obiges Beispiel dargestellt. Jede Entscheidung x_t^s kann mit einem Knoten identifiziert werden, die Gleichheitszeichen zwischen den Knoten deuten die entsprechenden Identitäten an.

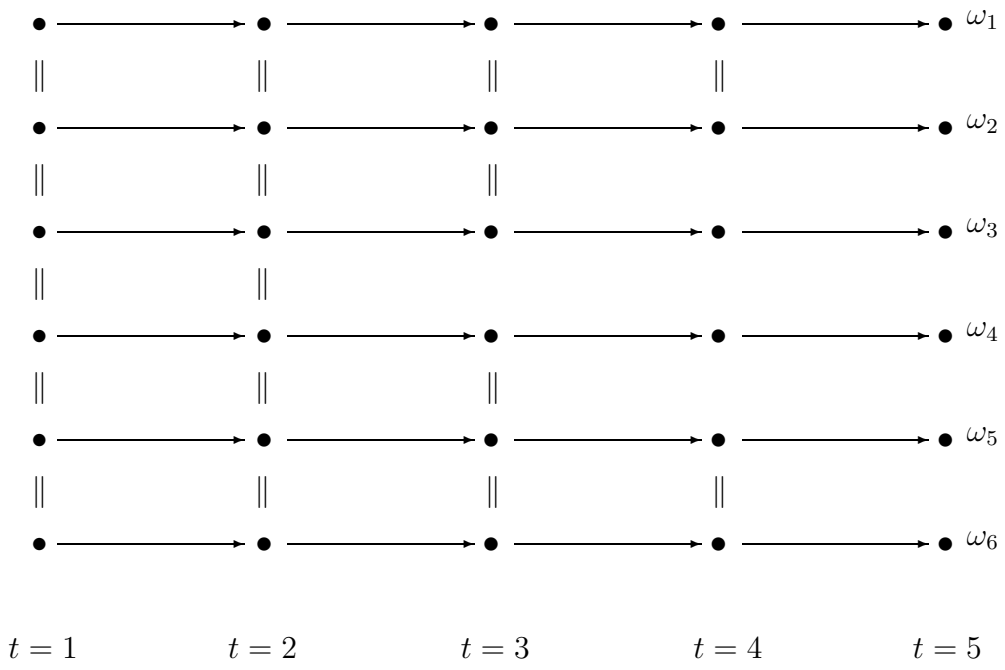


Abb. 4.2: Szenariobaum

Wir sind nun in der Lage unser erwartungswertbasiertes Problem (2.3) im diskreten Fall zu formulieren:

$$\min \left\{ \sum_{s=1}^{S} \pi_s \sum_{t=1}^{T} {c_t^s}^\top x_t^s \;\middle|\; \begin{array}{l} H_t(x_t) = 0,\ t = 1, \ldots, T, \\ \\ x_t^s \in X_t,\ s = 1, \ldots, S,\ t = 1, \ldots, T, \\ \\ \sum_{\tau=1}^{t} A_{t\tau}^s x_\tau^s = b_t^s,\ s = 1, \ldots, S,\ t = 1, \ldots, T \end{array} \right\}, \tag{3.5}$$

und erhalten ein endlichdimensionales gemischt-ganzzahliges Optimierungsproblem. Es sei an dieser Stelle bemerkt, dass die Formulierung eines mehrstufigen stochastischen Optimierungsproblems auf mehrere Weisen möglich ist; vergleiche beispielsweise Kapitel 3.5 in [18] oder Kapitel 2.4 in [97].

Wir wollen nun auf die Struktur der Nebenbedingungsmatrix von (3.5) eingehen und definieren dazu für ein $s \in \{1, \ldots, S\}$ folgende Matrix:

$$A^s := \begin{pmatrix} A_{11}^s & & & \\ A_{21}^s & A_{22}^s & & \\ \vdots & & \ddots & \\ A_{T1}^s & A_{T2}^s & \ldots & A_{TT}^s \end{pmatrix}$$

Die Dreiecksform resultiert aus der Tatsache, dass die Entscheidung in der ersten Stufe alle weiteren Zeitstufen beeinflussen kann; die Entscheidung in der zweiten Stufe die Zeitstufen zwei bis T usw.. Mit dieser Notation hat die Nebenbedingungsmatrix von (3.5) die Gestalt wie in Abbildung 4.3 dargestellt. Die Matrix H repräsentiert die aus den Nichtantizipativitätsbedingungen resultierenden Identitäten. Wir sehen dass diese Identitäten die einzigen Nebenbedingungen sind, die Entscheidungsvariablen verschiedener Szenarien miteinander koppeln. Relaxiert man diese Bedingungen, zerfällt das Problem in szenarioviele Teilprobleme, die dann unabhängig voneinander berechnet werden können.

Genau diese Dekompositionseigenschaft ist die Basis, auf die das in Kapitel 3.3 beschriebene Lösungsverfahren aufbaut.

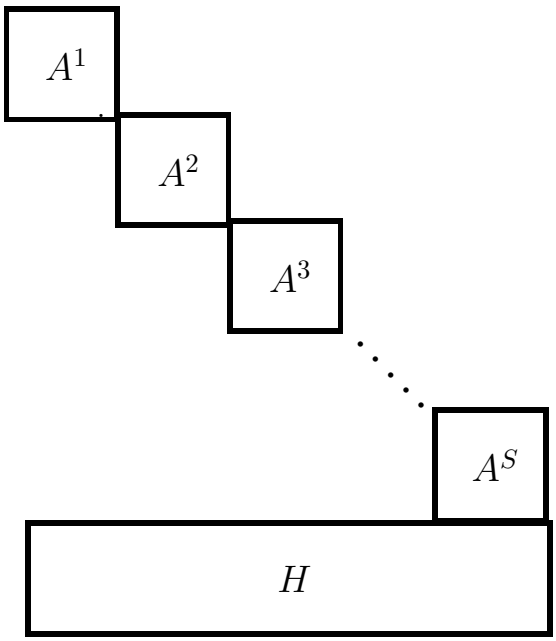


Abb. 4.3: Nebenbedingungsmatrix

3.2.1 Mean-Risk Modelle

Wie in Kapitel 2.3.2 festgestellt wurde, bleibt die Separabilität des Wait-and-See Problems in $\omega \in \Omega$ bei der Verwendung polyedrischer Risikomaße erhalten, vergleiche Lemma 2.17. Das heißt, die zusätzlichen Bedingungen zur Induzierung des Risikomaßes in (2.27), die im diskreten Fall für ein $s \in \{1, \ldots, S\}$ die Gestalt

$$\sum_{\tau=0}^{t-1} g_{t,\tau}^{\top} y_{t-\tau}^{s} + \sum_{\tau=1}^{t} {c_{\tau}^{s}}^{\top} x_{\tau}^{s} = 0, \ t = 1 \ldots, T \tag{3.6}$$

haben, führen zu keiner weiteren Kopplung von Variablen verschiedener Szenarien, so dass die Struktur der Nebenbedingungsmatrix in Abb. 4.3 erhalten bleibt. Nach Bemerkung 2.10 ist diese Aussage auch auf das Mean-Risk Modell, welches ein polyedrisches Risikomaß enthält, übertragbar.

Zwar gibt es für die Excess Probability keine bekannte Darstellung als polyedrisches Risikomaß, wie Beispiel 2.18 jedoch zeigte, ist das Problem ebenfalls separabel in $\omega \in \Omega$.

Betrachten wir nun ein Risikomaß, welches nicht separabel in $\omega \in \Omega$ ist. Nehmen

wir dazu die Semideviation aus Beispiel 2.20. Die Semideviation kann im diskreten Fall formuliert werden durch

$$\min \left\{ \sum_{t=2}^{T} \gamma_t \sum_{s=1}^{S} \eta_t^s \;\middle|\; \begin{array}{l} x \in \mathcal{X}(\xi), \\ \sum_{\tau=1}^{t} {c_\tau^s}^\top x_\tau^s - \sum_{s=1}^{S} \sum_{\tau=1}^{t} {c_\tau^s}^\top x_\tau^s \leq \eta_t^s, \;\; , s = 1, \ldots, S \\ \eta_t^s \in \mathbb{R}_+, \;\; H(\eta_t) = 0, \;\; t = 2, \ldots, T, \;\; s = 1, \ldots, S \end{array} \right\}. \quad (3.7)$$

Die Nebenbedingungsmatrix ist in Abbildung 4.4 dargestellt. Mit SD ist die Matrix mit den für die Semideviation zusätzlichen Nebenbedingungen gekennzeichnet. Vergleicht man Abbildung 4.3 mit Abbildung 4.4, so stellt sich die Frage, wieso man nicht sowohl die Nichtantizipativitätsbedingungen als auch die zusätzlichen Bedingungen für das Risikomaß relaxiert, um den gleichen Dekompositionseffekt zu erzielen.

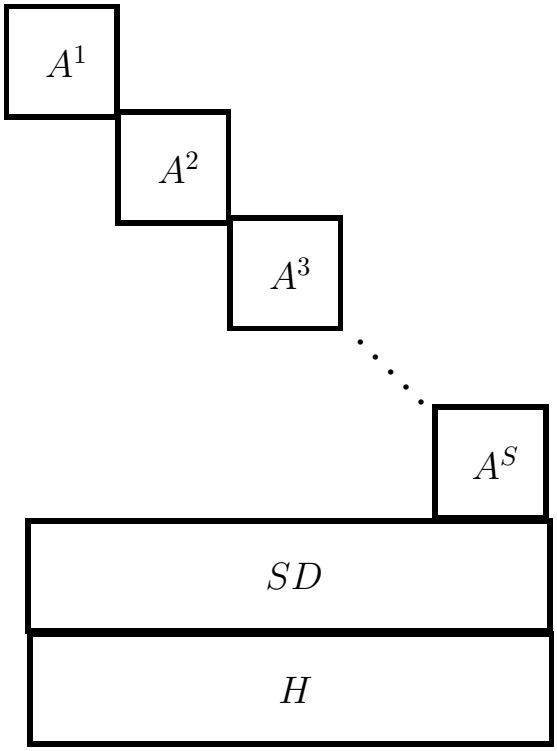


Abb. 4.4: Nebenbedingungsmatrix

Der Grund liegt darin, dass die Nichtantizipativitätsbedingungen einfache Gleich-

heitsbedingungen sind, siehe (3.4). Diese können relativ einfach "repariert" werden, während die Nebenbedingungen des Risikomaßes zum einen von komplexerer Natur sind und zum anderen die Anzahl dieser Nebenbedingungen identisch mit der Anzahl der Szenarien sind. Mehr dazu im folgenden Kapitel, wo ein sukkzesives Verfahren zur Wiederherstellung der Nichtantizipativitätsbedingungen vorgestellt wird.

Wir halten also fest, dass sowohl polyedrische Risikomaße wie das $\mathcal{R}_{M-CVaR}$ in Beispiel 2.24 oder das $\mathcal{R}_{M-EE}$ in Beispiel 2.15, als auch das in $\omega \in \Omega$ separabele Excess Probability $\mathcal{R}_{M-EE}$ die Struktur der Nebenbedingungsmatrix in Abbildung 4.3 erhalten.

3.3 Lösungsverfahren

In diesem Kapitel wird das grundsätzliche Lösungsverfahren vorgestellt, das beiden Algorithmen als Basis dient. Das Verfahren wird anhand des Erwartungswertmodells (3.5) erläutert und kann auch auf Risikomaßmodelle bzw. Mean-Risk Modelle angewandt werden, die die Struktur der Nebenbedingungsmatrix in Abbildung 4.3 erhalten, siehe Kapitel 3.2.1.

Folgende Ideen werden von beiden Algorithmen genutzt:

1. Relaxiere die Nichantizipativitätsbedingungen.

2. Initialisiere ein koordiniertes Branch-&-Bound Verfahren, das die relaxierten Bedingungen wiederherstellt.

3. Benutze ein Verwaltungssystem, um eine mehrfach Berechnung von Szenario-Teilproblemen zu vermeiden.

1. Relaxierung der Nichantizipativitätsbedingungen

Ohne Nichtantizipativitätsbedingungen zerfällt (3.5) in szenarioviele Teilprobleme,

siehe Abbildung 4.5.

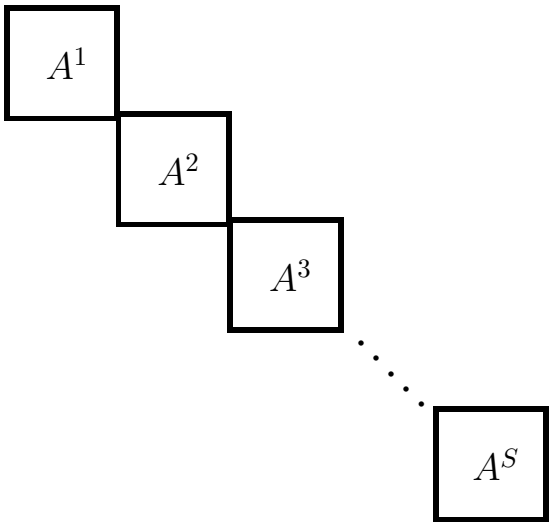


Abb. 4.5: Relaxierte Nebenbedingungsmatrix

Durch die Relaxierung geht Problem (3.5) über in

$$\min \left\{ \sum_{s=1}^{S} \pi_s \sum_{t=1}^{T} {c_t^s}^\top x_t^s \;\middle|\; \begin{array}{l} x_t^s \in X_t,\ s = 1, \ldots, S,\ t = 1, \ldots, T, \\ \\ \sum\limits_{\tau=1}^{t} A_{t\tau}^s x_\tau^s = b_t^s,\ s = 1, \ldots, S,\ t = 1, \ldots, T \end{array} \right\} \tag{3.8}$$

Da die NA-Bedingungen die einzigen Nebenbedingungen sind, die verschiedene Szenarien miteinander koppeln, kann Erwartungswert und Minimum vertauscht werden und man erhält

$$\sum_{s=1}^{S} \pi_s \min \left\{ \sum_{t=1}^{T} {c_t^s}^\top x_t^s \;\middle|\; \begin{array}{l} x_t^s \in X_t,\ t = 1, \ldots, T, \\ \\ \sum\limits_{\tau=1}^{t} A_{t\tau}^s x_\tau^s = b_t^s\ t = 1, \ldots, T \end{array} \right\}. \tag{3.9}$$

Diese Teilprobleme können nun unabhängig voneinander berechnet werden und werden im Weiteren als Szenarioprobleme bezeichnet.

2. Koordiniertes Branch-&-Bound Verfahren

Um die relaxierten Bedingungen wieder herzustellen gibt es eine Reihe von Methoden, siehe etwa [5, 6, 20, 54]. In [20] nutzt man beispielsweise für den zweistufigen Fall, das Lagrange'sche Dualproblem, welches zu einem nicht-linearen konkaven Maximierungsproblem führt. Die Dimension des Raumes in dem das Dualproblem optimiert wird, ist gleich der Anzahl der Nichtantizipativitätsbedingungen. Da im mehrstufigen Fall die Anzahl der Nichtantizipativitätsbedingung stark zunimmt und das duale Lagrange Problem durch die hohe Dimension nur sehr schwer zu lösen ist, wird die folgende Methode angewandt.

Wir werden hier das koordiniertes Branch-&-Bound Verfahren nutzen, wie es erstmals in [5, 6] vorgestellt wurde. Dazu folgende Definition

Definition 3.1 (Familien von Variablen)
Für ein festes $t \in \{1, \ldots, T\}$ *definieren wir eine Familie von Variablen* $\mathcal{G}_t$ *wie folgt: Zwei Variablen* $x_t^s, x_t^{s'}$ *mit* $s, s' \in \{1, \ldots, S\}$ *gehören zur selben Familie* $\mathcal{G}_t$ *genau dann wenn* $\exists E \in \mathcal{E}_t : \omega_s, \omega_{s'} \in E$.

Wie in Kapitel 3.2 bereits dargestellt sind die NA-Bedingungen einfache Gleichheitsbedingungen, vergleiche (3.4). Definition 3.1 bedeutet anschaulich, dass Variablen die aufgrund dieser Gleichheitsbedingungen identisch sein sollen, zu Familien zusammengefaßt werden. Betrachten wir dazu den Szenariobaum in Abbildung 4.6. Es ist ein Szenariobaum mit drei Zeitstufen und vier Szenarien dargestellt. Die Gleichheitszeichen im rechten Bild deuten die Nichtantizipativitätsbedingungen an. Wir haben also die folgenden Familien:

$$t = 1 : \mathcal{G}_1^1 := \{x_1^1, x_1^2, x_1^3, x_1^4\} \qquad t = 2 : \mathcal{G}_2^1 := \{x_2^1, x_2^2\} \quad \mathcal{G}_2^2 := \{x_2^3, x_2^4\}$$

$$t = 3 : \mathcal{G}_3^1 := \{x_3^1\} \quad \mathcal{G}_3^2 := \{x_3^2\} \quad \mathcal{G}_3^4 := \{x_3^3\} \quad \mathcal{G}_3^5 := \{x_3^4\}$$

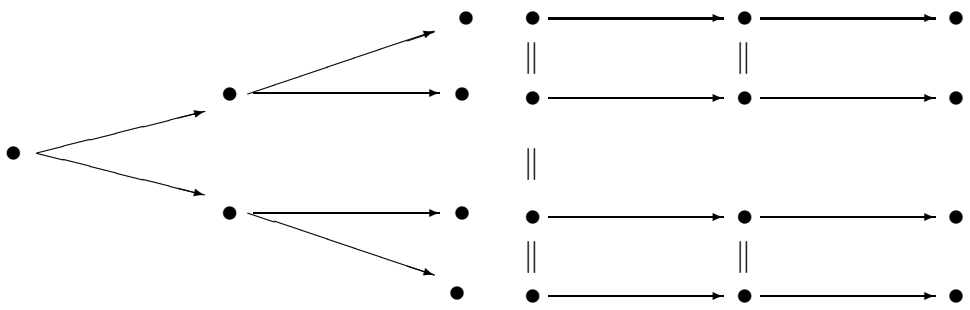


Abbildung 4.6: Szenariobaum

Um die relaxierten Bedingungen wieder herzustellen werden alle Variablen, die zur selben Familie gehören simultan gebrancht. Das Verfahren soll anhand Abbildung 4.7 erläutert werden, wobei der Informationsverlauf aus Abbildung 4.6 mit den entsprechenden Familien betrachtet wird. Um eine klare Begriffsabgrenzung zu bekommen, bezeichnen wir den Baum in Abbildung 4.7 als Masterbaum, die Knoten als Masterknoten M, die die Gesamtprobleme repräsentieren. Diese wiederum bestehen aus szenariovielen Szenarioproblemen $P_M^1, \ldots, P_M^S$.
Der Wurzelknoten repräsentiert das Ausgangsproblem (3.9), welches aus den Szenarioproblemen 1.1, 2.1, 3.1, 4.1 besteht. Der Einfachheit halber seien alle Variablen binär. Im Wurzelknoten wird die Familie $\mathcal{G}_2^2$ gebrancht. Alle Variablen dieser Familie werden entweder gleich Null oder gleich Eins gesetzt. Um den Wert des linken Masterproblems zu erhalten, werden die Szenarioprobleme 1.2, 2.2, 3.2, 4.2 berechnet, für den Wert des rechten Masterproblems werden die Szenarioprobleme 1.3, 2.3, 3.3, 4.3 berechnet. Im folgenden Schritt wird dann in den zwei aktiven Masterknoten $\{1.2,2.2,3.2,4.2\}$ und $\{1.3,2.3,3.3,4.3\}$ die Familie $\mathcal{G}_2^1$ gebrancht und man setzt wiederum alle Variablen dieser Familie entweder gleich Null oder gleich Eins. Um die Werte der daraus resultierenden Masterknoten zu erhalten, werden wiederum die entsprechenden neu entstandenen Szenarioprobleme berechnet.

Wie beim Branchen zur Wiederherstellung der Ganzzahligkeit, liefert das Verfahren untere Schranken für das Gesamtproblem. Obere Schranken können mit Hilfe von Heuristiken ermittelt werden. Es hat sich hierbei als sinnvoll erwiesen, die Heu-

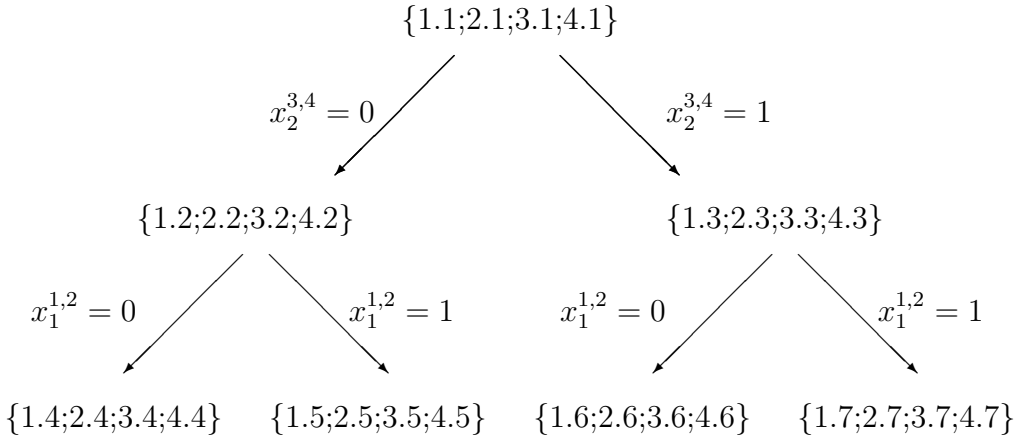


Abb. 4.7: Masterbaum

ristiken dem Informationsverlaufes anzupassen. Die Variablen werden in den einzelnen Zeitschritten nach und nach fixiert. Das heißt, fixiere die Variablen in der Zeitstufe $t \in \{1, \ldots, T\}$ und berechne das relaxierte Problem. Falls dieses zulässig ist, fixiere die Variablen einer anderen Zeitstufe $t' \in \{1, \ldots, T\}$, $t' \neq t$ und berechne das relaxierte Problem, usw..
Wir haben nun das folgende algorithmische Basisschema:

SCHRITT 1: (Initialisierung):
Setze Masterproblem (3.9) auf Masterliste $\mathbf{M}$ und die entsprechenden Szenarioprobleme in eine Szenarioliste $\mathbf{P}$. Setze z:=$+\infty$.

SCHRITT 2: (Knotenauswahl):
Falls $\mathbf{M} = \emptyset$, TERMINATE. Falls nicht, wähle nächstes Masterproblem $M \in \mathbf{M}$ und die zugehörigen Szenarioprobleme $P_M^1, \ldots, P_M^S \in \mathbf{P}$. Lösche M aus Liste.

SCHRITT 3: (Berechnung):
Berechne die Szenarioprobleme.

SCHRITT 4: (Zulässigkeit):
Falls ein P_M^s, $s \in \{1, \ldots, S\}$ unzulässig ist, lösche $M \in \mathbf{M}$ und alle Szenarioprobleme

$P_M^1, \ldots, P_M^S \in \mathbf{P}$ und gehe zu SCHRITT2. Andernfalls berechne den Zielfunktionswert $\varphi(M)$ zur Lösung $x(M)$.

SCHRITT 5: (Inferiorität):
Falls z $\leq \varphi(M)$, gehe zu SCHRITT 2.

SCHRITT 6: (Optimalität):
Falls $x(M)$ die relaxierten Bedingungen erfüllt, setze z $= \varphi(M)$, lösche alle $M' \in \mathbf{M}$ und die entsprechenden Szenarioprobleme in $\mathbf{P}$ falls LB(M') $\geq \varphi(M)$, gehe zu SCHRITT2.

SCHRITT 7: (Heuristik):
Versuche mittels Heuristiken, aus $x(M)$ eine zulässige Lösung mit Zielfunktionswert $h(M)$ zu generieren. Falls $h(M) < z$, setze $z = h(M)$, lösche alle $M' \in \mathbf{M}$ und die entsprechenden Szenarioprobleme in $\mathbf{P}$ falls LB(M') $\geq h(M)$.

SCHRITT 8: (Branching):
Treffe eine Branchingentscheidung. Es sei $\mathcal{G}_t^r$ die zu branchende Familie mit dem Index (i). Erzeuge für alle $P_M^s,\ s \in \{1, \ldots, S\}$ zwei neue Probleme $P1^s, P2^s, s \in \{1, \ldots, S\}$ und füge die zusätzlichen Bedingungen:

$$x_{t,(i)}^s \leq \lfloor x_B \rfloor \text{ falls } x_{t,(i)}^s \text{ ganzzahlig},\ x_{t,(i)}^s \leq x_B - \gamma \text{ sonst},\ s \in \mathcal{G}_t^r$$

bzw

$$x_{t,(i)}^s \geq \lceil x_B \rceil \text{ falls } x_{t,(i)}^s \text{ ganzzahlig},\ x_{t,(i)}^s \leq x_B + \gamma \text{ sonst},\ s \in \mathcal{G}_t^r$$

für ein gegebenes $\gamma > 0$ hinzu, falls $s \in \mathcal{G}_t^r$. Erzeuge die entsprechenden Masterknoten im Masterbaum und gehe zu SCHRITT 2.

Der in Schritt 8 definierte Parameter γ soll garantieren, dass man disjunkte Teilprobleme beim Branchen auf stetigen Variablen bekommt.

Bei den Branchingentscheidungen in Schritt 8 ist es möglich eine konkrete Prioritätsordnung vorzugeben. Außerdem wurden die folgenden Regeln implementiert:
- Wähle die Variable mit dem kleinsten Zeitindex $t \in \{1, \ldots, T\}$,

- Ziehe ganzzahlige Variablen vor.

Beide Regeln können miteinander kombiniert werden.

Für die Auswahl des Masterknotens in Schritt 2 wurde eine "kleinste untere Schranke Strategie" in beiden Algorithmen implementiert: Wähle das Masterproblem mit der kleinsten unteren Schranke.

Bei praktischen Anwendungen verlangt oft die Größe der Probleme ein Abbruchkriterium, wie eine Beschränkung der Rechenzeit oder eine vorgegebene Toleranz zwischen oberer und unterer Schranke, die bei Unterschreitung zum Abbruch führt. Ein Endlichkeitskriterium liefert die folgende Aussage.

Lemma 3.2 *Unter der Voraussetzung, dass* $X := X_1, \times X_2 \times, \ldots, \times X_T$, *in (3.5) beschränkt ist, terminiert obiger Algorithmus nach endlichen vielen Schritten.*

Beweis

Da X beschränkt ist, existieren zu jeder Komponente $x^s_{t,(i)}$ von $x \in X$ eine größte untere Schranke $l^s_{t,(i)}$ und eine kleinste obere Schranke $u^s_{t,(i)}$ mit $l^s_{t,(i)} \leq x^s_{t,(i)} \leq u^s_{t,(i)}$. Durch das Branchen in Schritt 8 des Algorithmus, wird X in disjunkte Partitionen unterteilt. Bei ganzzahligen Komponenten gibt es nur endlich viele zulässige Werte. Durch den konstanten Parameter $\gamma > 0$ wird eine endlose Verfeinerung dieser Partitionen bei stetigen Variablen verhindert. D.h., ein Teilproblem mit der zulässigen Menge $X' \in X$ wird nicht weiter unterteilt, falls für alle ganzzahligen Komponenten

$$|{u^s_{t,(i)}}' - {l^s_{t,(i)}}'| < 1$$

und alle stetigen Komponenten

$$|{u^s_{t,(i)}}' - {l'^s_{t,(i)}}'| < \gamma$$

gilt. Dadurch kann es nur zur einer endlichen Anzahl an Teilproblemen kommen, die berechnet werden müssen. □

3. Entwurf eines Verwaltungssystems

Wir kommen nun zu dem Teil des Lösungsverfahrens, welches die beiden Algorithmen bb-tree und bb-hash voneinander unterscheidet. In dem oben beschriebenen

Basisverfahren werden zu jedem neu erzeugten Masterknoten genau szenarioviele Szenarioprobleme berechnet. Aufgrund der Tatsache jedoch, dass nicht alle Familien von Variablen ein Element aus jedem Szenario enthalten, können einige Szenarioprobleme in mehreren Masterknoten verwendet werden.

Schauen wir uns noch einmal den Masterbaum in Abbildung 4.7 an. Als im Wurzelknoten {1.1;2.1;3.1;4.1} die Familie $\mathcal{G}_2^2$ zum Branchen gewählt wurde, betraf diese Entscheidung nur die Szenarien 3 und 4. Somit wurden nur die Szenarioprobleme 3.2, 4.2 bzw. 3.3, 4.3 in Schritt 7 des Algorithmus manipuliert. Die Szenarioprobleme 1.1, 2.1 sind mit den Szenarioproblemen 1.2, 2.2 und 1.3, 2.3 identisch. Dadurch entstehen Einspareffekte beim Berechnen der Szenarioprobleme, denn zur Erzeugung des Masterknotens {1.2;2.2;3.2;4.2} müssen nur die Szenarioprobleme 3.2 und 4.2 neu berechnet werden. Gleiches gilt für den Masterknoten {1.3;2.3;3.3;4.3} bei dem nur die Szenarioprobleme 3.3 und 4.3 neu berechnet werden müssen.

Genau um diese Einspareffekte geht es bei den entwickelten Verwaltungssystemen. Es werden nun zwei unterschiedliche Verwaltungssysteme vorgestellt die zu den Algorithmen bb-tree und bb-hash führen.

3.3.1 bb-tree

Bei diesem Verwaltungssystem werden die Szenarioprobleme getrennt voneinander in szenarioviele Listen $\mathbf{P^1}, \ldots, \mathbf{P^S}$ mit binärer Baumstruktur gespeichert, siehe Abbildung 4.8. Diese sollen im Weiteren als Szenario-Branching-Bäume (SBB) bezeichnet werden, die Knoten als Szenarioknoten. In diesem Beispiel hat das Gesamtproblem vier Szenarien. Die Notation wurde aus dem vorigen Kapitel übernommen. Der Wurzelknoten des ersten SBB entspricht dem ersten Szenarioproblem von (3.9). Entsprechendes gilt für die anderen SBB. Wenn eine Branchingentscheidung getroffen wurde, wird in allen SBB die von dieser Entscheidung betroffen sind gebrancht.

Obiges Beispiel wird hier noch einmal aufgenommen, siehe Abbildung 4.6. Der neu entstandene Masterbaum ist in Abbildung 4.9 zu sehen. Im Wurzelknoten des Masterbaumes wird die Familie $\mathcal{G}_2^2$ gebrancht. Diese Entscheidung betrifft die SBB der Szenarien 1 und 2. In beiden SBB wird entsprechend die Entscheidung ausgeführt. Im Masterbaum werden die Szenarioprobleme 1.1 und 2.1 beim Erzeugen

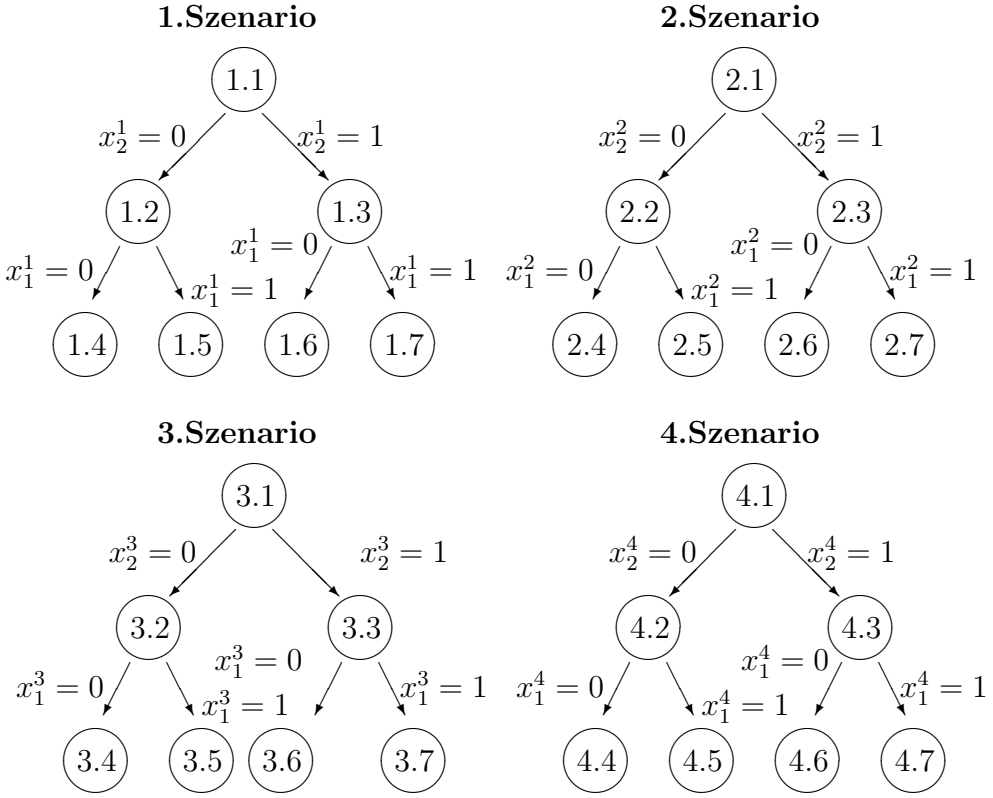


Abb. 4.8: SBB

der Nachfolgeknoten aus dem Vorgängerknoten übernommen. Es sind nur zwei neue Szenarioprobleme zu berechnen. Vergleicht man die Masterknoten in Abbildung 4.7 und Abbildung 4.9, wird das Einsparpotential an Szenarioknoten klar.

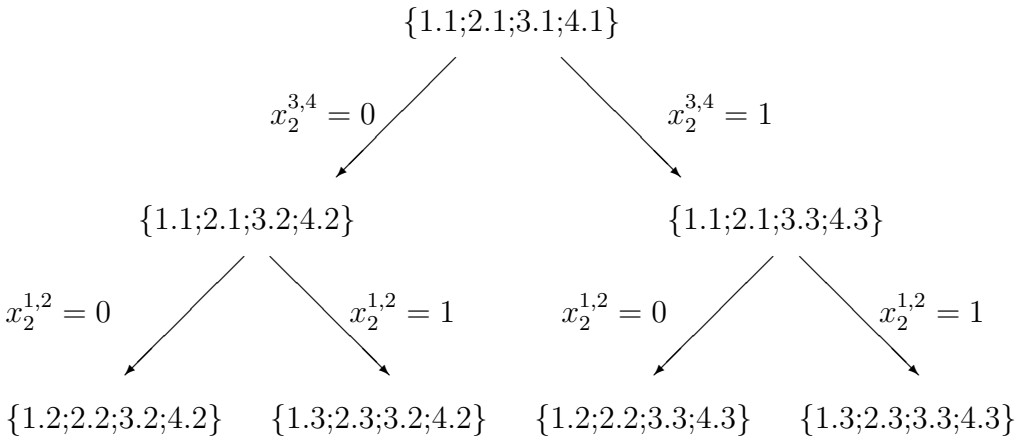


Abb. 4.9: Masterbaum

Bei diesem Verwaltungsverfahren muss allerdings dafür Sorge getragen werden, dass die binäre Struktur der SBB erhalten bleibt. Würde man in Abbildung 4.9 auf die Masterknoten {1.1;2.1;3.2;4.2} und {1.1;2.1;3.3;4.3} verschiedene Branchingentscheidungen anwenden, so wäre es möglich, dass man den Szenarioknoten, die in diesen Masterknoten enthalten sind, mehrere Nachfolger zuordnen muss. Wir fügen deshalb eine Branchingliste **B** ein. Angefangen mit der Ebene Null, die nur den Wurzelknoten des Baumes enthält, seien mit der i-ten Ebene alle Knoten des Masterbaumes bezeichnet, die zur Erzeugung $i \in \mathbb{N}$ Branchingentscheidungen benötigt. In Abbildung 4.9 sehen wir ein Masterbaum mit drei Ebenen. Indem wir in jeder Ebene des Masterbaumes dieselbe Branchingstrategie ausüben, garantieren wir die binäre Struktur der SBB.

Des Weiteren muss festgelegt werden, unter welchen Bedingungen ein Szenarioknoten in einem SBB gelöscht werden kann. Die Tatsache, dass ein Szenarioknoten in keinem aktiven Masterknoten erhalten ist, reicht nicht aus, da ein neu generierter Masterknoten wiederum auf einen bereits berechneten Szenarioknoten zurückgreifen könnte. Es muss daher folgendes gegeben sein.

Lemma 3.3 *Ein Szenarioknoten $P^s \in \mathbf{P}^s$, $s \in \{1, \ldots, S\}$ wird im weiteren Verlauf des Algorithmus nicht erneut erzeugt, falls sein Vorgängerknoten bereits gelöscht wurde. Er kann in dem Fall ebenfalls gelöscht werden, falls er in keinem aktiven Masterknoten $M \in \mathbf{M}$ enthalten ist.*

Beweis

Der Satz erschließt sich induktiv. Neue Szenarioknoten werden ausschließlich als Nachfolger von zu branchenden Szenarioknoten erzeugt. Somit gilt die Aussage offenbar für die Wurzelknoten der SBB, da sie keine Vorgänger haben.

Es sei $P_0^{s_0} \in \mathbf{P}^{s_0}$, $s_0 \in \{1, \ldots, S\}$ ein Szenarioknoten für den die Aussage gelte und $P_1^{s_0} \in \mathbf{P}^{s_0}$ ein Nachfolgerknoten. Falls $P_0^{s_0}$ glöscht wird, wird er nicht erneut erzeugt. Mit obigen Argument wird dann $P_1^{s_0}$ ebenfalls nicht mehr neu erzeugt und kann gelöscht werden, wenn er in keinem aktiven Masterknoten enthalten ist. □

'bb-tree'

SCHRITT 1: (Initialisierung):
Setze Masterproblem M mit Ebene(M):=0 auf Masterliste $\mathbf{M}$ und Szenarioprobleme auf die entsprechenden Szenariolisten $\mathbf{P}^s$, $s = 1, \ldots, S$. Initialisiere Branchingliste $\mathbf{B}:= \emptyset$, setze z:=$+\infty$ und MAXTIEFE:=0.

SCHRITT 2: (Knotenauswahl):
Falls $\mathbf{M} = \emptyset$, TERMINATE. Falls nicht, wähle nächstes Masterproblem $M \in \mathbf{M}$ und die zugehörigen Szenarioprobleme $P_M^1 \in \mathbf{P}^1, \ldots, P_M^S \in \mathbf{P}^S$. Lösche M aus Liste.

SCHRITT 3: (Berechnung):
Berechne alle nicht bereits berechneten Szenarioprobleme.

SCHRITT 4: (Zulässigkeit):
Falls gewisse P_M^s unzulässig sind, lösche diese Probleme aus den entsprechenden Szenariolisten, lösche alle $M' \in \mathbf{M}$, welche die entsprechenden Szenarioprobleme enthalten, aktualisiere die übrigen Szenarioprobleme und gehe zu SCHRITT 2. Andernfalls berechne den Zielfunktionswert $\varphi(M)$ zur Lösung $x(M)$.

SCHRITT 5: (Inferiorität):
Falls z $\leq \varphi(M)$, aktualisiere die Szenarioprobleme und gehe zu SCHRITT 2.

SCHRITT 6: (Optimalität):
Falls $x(M)$ die relaxierten Bedingungen erfüllt, setze z = $\varphi(M)$, lösche alle $M' \in \mathbf{M}$ mit LB(M') $\geq \varphi(M)$, aktualisiere die Szenarioprobleme und gehe zu SCHRITT 2.

SCHRITT 7: (Heuristik):
Versuche mittels Heuristiken, aus $x(M)$ eine zulässige Lösung mit Zielfunktionswert $h(M)$ zu generieren. Falls $h(M) < z$, setze $z = h(M)$, lösche alle $M' \in \mathbf{M}$ mit LB(M') $\geq h(M)$ und aktualisiere die Szenarioprobleme.

SCHRITT 8: (Branching):
i) Falls Ebene(M)=MAXTIEFE, erzeuge in Branchingliste **B** eine neues Element B mit Ebene(B)=MAXTIEFE, treffe eine Branchingentscheidung und speichere diese in B. Setze MAXTIEFE=MAXTIEFE+1, erzeuge bezüglich dieser getroffenen Entscheidung die entsprechenden Szenarioprobleme in den Szenariolisten* und füge der Masterliste **M** zwei neue Probleme M1 und M2 mit
Ebene(M1)=Ebene(M2)=Ebene(M)+1 hinzu, gehe zu SCHRITT 2.
ii) Falls Ebene(M) < MAXTIEFE, übernehme die Branchingentscheidung $B \in \mathbf{B}$ mit Ebene(B)=Ebene(M), erzeuge die entsprechenden Szenarioprobleme in den Szenariolisten und füge der Masterliste **M** zwei neue Probleme M1 und M2 mit Ebene(M1)=Ebene(M2)=Ebene(M)+1 hinzu, gehe zu SCHRITT 2.

* Es sei $\mathcal{G}_t^r$ die zu branchende Familie mit dem Index (i). Erzeuge für alle $P_M^s,\ s \in \mathcal{G}_t^r$ zwei neue Probleme $P1^s, P2^s, s \in \mathcal{G}_t^r$ mit den zusätzlichen Bedingungen:

$$x_{t,(i)}^s \leq \lfloor x_B \rfloor \text{ falls } x_{t,(i)}^s \text{ ganzzahlig},\ x_{t,(i)}^s \leq x_B - \gamma \text{ sonst},\ s \in \mathcal{G}_t^r$$

bzw

$$x_{t,(i)}^s \geq \lceil x_B \rceil \text{ falls } x_{t,(i)}^s \text{ ganzzahlig},\ x_{t,(i)}^s \leq x_B + \gamma \text{ sonst},\ s \in \mathcal{G}_t^r$$

für ein gegebenes $\gamma > 0$, und setze diese auf die entsprechenden Szenariolisten $\mathbf{P}^s, s \in \mathcal{G}_t^r$.

Wir werden diesen Algorithmus anhand eines kleinen Beispiels demonstrieren:

Beispiel 3.4 *Es sei ein Informationsverlauf gegeben, wie in Abbildung 4.6 dargestellt. Nur die rechte Seite b sei zufallsbehaftet und genüge folgender Verteilung:*

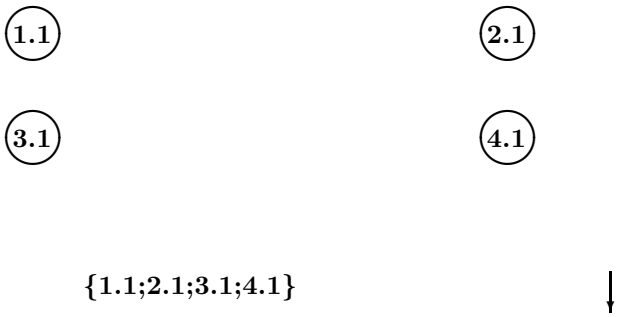
$$\begin{array}{lll}
\mathbb{P}(\{\omega_1\}) = 0.25 & b_1(\{\omega_1\}) = 0 & b_2(\{\omega_1\}) = 0 \\
\mathbb{P}(\{\omega_2\}) = 0.25 & b_1(\{\omega_2\}) = 0 & b_2(\{\omega_2\}) = 1 \\
\mathbb{P}(\{\omega_3\}) = 0.25 & b_1(\{\omega_3\}) = 1 & b_2(\{\omega_3\}) = 0 \\
\mathbb{P}(\{\omega_4\}) = 0.25 & b_1(\{\omega_4\}) = 1 & b_2(\{\omega_4\}) = 1
\end{array}$$

Die restlichen Daten sind:

$$\min \left\{ \sum_{s=1}^{4} 0.25(-x_1^s - x_2^s + 5y_2^s + 1.5y_3^s) \;\middle|\; \begin{array}{l} x_1^1 = x_1^2 = x_1^3 = x_1^4, \\ x_2^1 = x_2^2,\ x_2^3 = x_2^4,\ y_2^1 = y_2^2,\ y_2^3 = y_2^4, \\ \\ x_1^s, x_2^s \in \{0,1\},\ \ y_2^s, y_3^s \in \mathbb{Z}_+, \\ s = 1, \ldots, 4, \\ \\ x_1^s \leq b_1^s + y_2^s,\ \ x_1^s + x_2^s \leq b_2^s + y_3^s, \\ s = 1, \ldots, 4 \end{array} \right\}$$

Iteration 1:

Die oberen Knoten stellen die Wurzelknoten der SBB's dar. Wie in obiger Notation ist hier der Wurzelknoten des Masterbaumes dargestellt. Der Pfeil deutet die Branchingliste an, die zu Beginn leer ist.



1. Schritt:

Setze $z = -\infty$, $MAXTIEFE = 0$,

$\mathbf{M} = \{\{1.1, 2.1, 3.1, 4.1\}\}$,

$\mathbf{P}^1 = \{(1.1)\}$, $\mathbf{P}^2 = \{(2.1)\}$, $\mathbf{P}^3 = \{(3.1)\}$, $\mathbf{P}^4 = \{(4.1)\}$.

2. Schritt:

Wähle den Masterknoten $\{1.1, 2.1, 3.1, 4.1\} \in \mathbf{M}$.

3. Schritt:

Berechne die Szenarioprobleme $(1.1), (2.1), (3.1), (4.1)$ und erhalte die Lösung

$$y_3^1 = 0,$$
$$x_2^1 = 0,\ x_2^2 = 1,$$
$$y_2^1 = 0,\ y_2^2 = 0,$$
$$y_3^2 = 0,$$
$$x_1^1 = 0,\ x_1^2 = 0,\ x_1^3 = 0,\ x_1^4 = 1,$$
$$y_3^3 = 0,$$
$$x_2^3 = 0,\ x_2^4 = 0,$$
$$y_2^3 = 0,\ y_2^4 = 0,$$
$$y_3^4 = 0,$$

mit dem Zielfunktionswert $\varphi(\{1.1, 2.1, 3.1, 4.1\}) = -0.5$.

4. Schritt:
Alle Szenarioprobleme sind zulässig.

5. Schritt:
Inferiorität ist nicht gegeben.

6. Schritt:
Optimalität ist nicht gegeben.

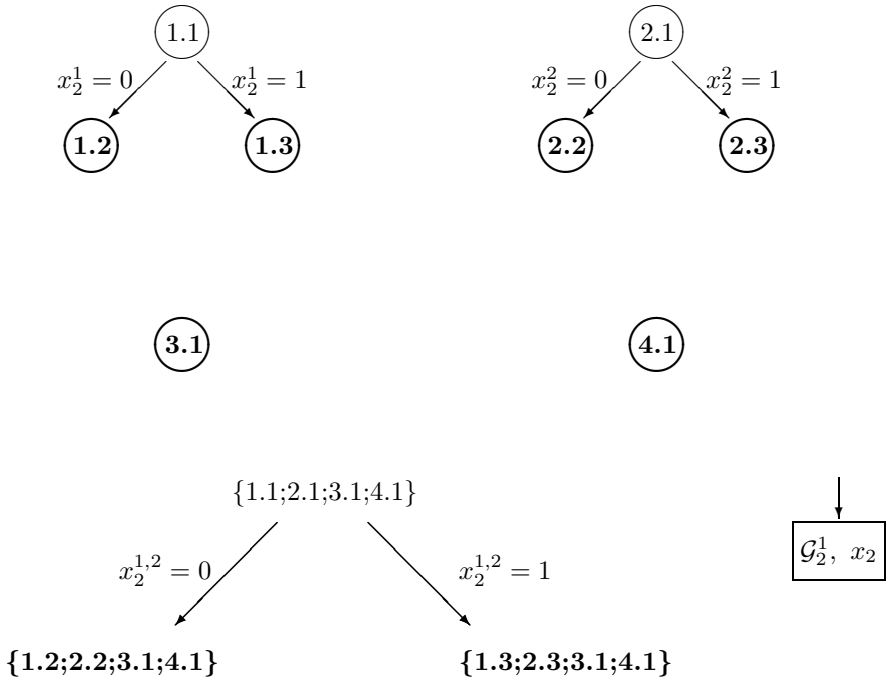
7. Schritt:
Erhalte aus der Heuristik die zulässige Lösung

$$y_3^1 = 1,$$
$$x_2^1 = 1,\ x_2^2 = 1,$$
$$y_2^1 = 0,\ y_2^2 = 0,$$
$$y_3^2 = 0,$$
$$x_1^1 = 0,\ x_1^2 = 0,\ x_1^3 = 0,\ x_1^4 = 0,$$
$$y_3^3 = 1,$$
$$x_2^3 = 1,\ x_2^4 = 1,$$
$$y_2^3 = 0,\ y_2^4 = 0,$$
$$y_3^4 = 0,$$

mit dem Zielfunktionswert h({1.1,2.1,3.1,4.1})=-0.25. Setze z=-0.25.

8.Schritt:
Ebene({1.1,2.1,3.1,4.1})=MAXTIEFE → Fall (i).
Setze MAXTIEFE=1. Wähle Familie $\mathcal{G}_2^1$ und Variable x_2.

Iteration 2:
Eine Branchingentscheidung wurde getroffen und in der Branchingliste gespeichert. Entsprechend wurden die Szenarioknoten und Masterknoten erzeugt. Die fett markierten Knoten sind aktive Knoten. Im Masterbaum wurde der Wurzelknoten gemäß des üblichen Branch-and-Bound Schemas gelöscht. Die Szenarioknoten im ersten und zweiten SBB wurden gelöscht, da sie als Wurzelknoten keinen Vorgänger haben und in keinem aktiven Masterknoten vorhanden sind. Wir sehen hier zum ersten Mal den Effekt der Wiederverwendung von Szenarioknoten; denn zum Erzeugen der Masterknoten $\{1.2, 2.2, 3.1, 4.1\}$ und $\{1.3, 2.3, 3.1, 4.1\}$ können die bereits berechneten Szenarioknoten (3.1) und (4.1) wieder verwendet werden.



2. Schritt:
Wähle den Masterknoten $\{1.3, 2.3, 3.1, 4.1\} \in \mathbf{M}$.

3. Schritt:
Berechne die Szenarioprobleme $(1.3), (2.3)$, übernehme $(3.1), (4.1)$ und erhalte die Lösung

$$\begin{array}{llll} & & & y_3^1 = 1, \\ & & x_2^1 = 1,\ x_2^2 = 1, & \\ & & y_2^1 = 0,\ y_2^2 = 0, & \\ & & & y_3^2 = 0, \\ x_1^1 = 0,\ x_1^2 = 0,\ x_1^3 = 0,\ x_1^4 = 1, & & & \\ & & & y_3^3 = 0, \\ & & x_2^3 = 0,\ x_2^4 = 0, & \\ & & y_2^3 = 0,\ y_2^4 = 0, & \\ & & & y_3^4 = 0, \end{array}$$

mit dem Zielfunktionswert $\varphi(\{1.1, 2.1, 3.1, 4.1\}) = -0.375$.

4. Schritt:
Alle Szenarioprobleme sind zulässig.

5. Schritt:
Inferiorität ist nicht gegeben

6. Schritt:
Optimalität ist nicht gegeben

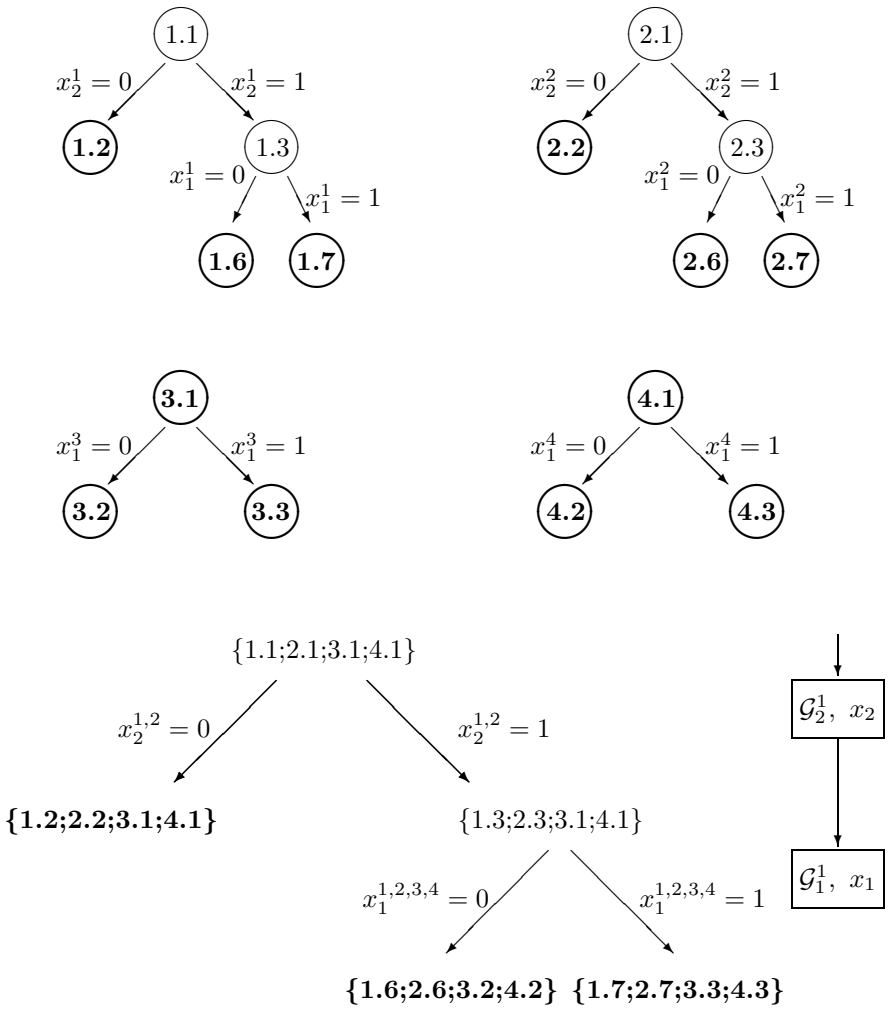
7. Schritt:
Erhalte keine zulässige Lösung.

8.Schritt:

Ebene({1.3,2.3,3.1,4.1})=MAXTIEFE → Fall (i). Setze MAXTIEFE=2. Wähle Familie $\mathcal{G}_1^1$ und Variable x_1.

Iteration 3:

Es wurde eine neue Branchingentscheidung getroffen und in der Branchingliste gespeichert. Zwei neue Masterknoten wurden erzeugt. Die zur Erzeugung der beiden Masterknoten nötigen Szenarioknoten wurden erzeugt. Gelöscht wurden die Szenarioknoten (1.3) und (2.3), da deren Vorgänger bereits gelöscht wurde und sie in keinem aktiven Masterknoten vorhanden sind. Einspareffekte gab es bei der Erzeugung der neuen Masterknoten nicht, da eine Erststufenvariable gebrancht wurde, bei der alle Szenarien involviert sind.



2. Schritt:
Wähle den Masterknoten $\{1.7; 2.7; 3.3; 4.3\} \in \mathbf{M}$.

3. Schritt:
Berechne die Szenarioprobleme $(1.7), (2.7), (3.3), (4.3)$ und erhalte die Lösung

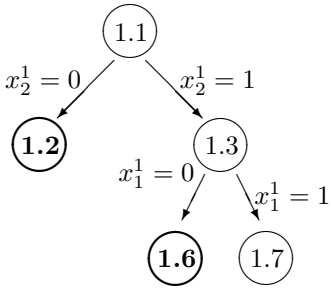
$$\begin{array}{llll} & & & y_3^1 = 1, \\ & & x_2^1 = 0,\ x_2^2 = 0, & \\ & & y_2^1 = 1,\ y_2^2 = 1, & \\ & & & y_3^2 = 0, \\ x_1^1 = 1,\ x_1^2 = 1,\ x_1^3 = 1,\ x_1^4 = 1, & & & \\ & & & y_3^3 = 1, \\ & & x_2^3 = 1,\ x_2^4 = 1, & \\ & & y_2^3 = 0,\ y_2^4 = 0, & \\ & & & y_3^4 = 0, \end{array}$$

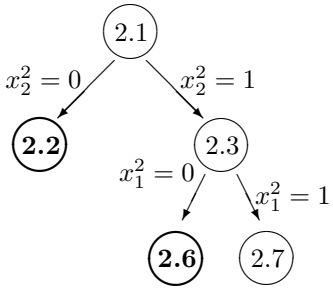
mit dem Zielfunktionswert $\varphi(\{1.7, 2.7, 3.3, 4.3\}) = 2.5$

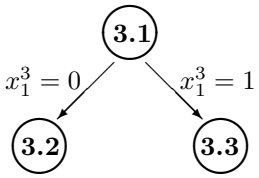
4. Schritt:
Alle Szenarioprobleme sind zulässig.

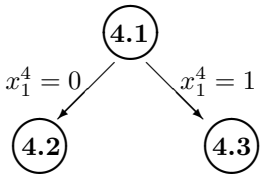
5. Schritt:
Inferiorität ist gegeben $\rightarrow$ gehe zu Schritt 2.

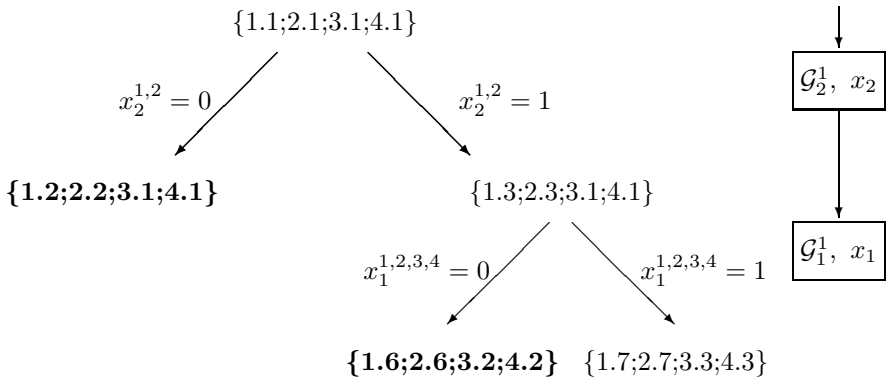
Iteration 4:
Der unterlegene Masterknoten wurde entsprechend des Branch-and-Bound Schemas gelöscht. Wir sehen hier zum ersten Mal, dass Szenarioknoten zwar nicht mehr in einem Masterknoten enthalten sind, diese aber nicht abgeschnitten werden können, da ihr Vorgänger noch nicht gelöscht wurde. Somit werden nur die Szenarioknoten (1.7) und (2.7) abgeschnitten, nicht jedoch die Knoten (3.3) und (4.3).











2. Schritt:

Wähle den Masterknoten $\{1.2; 2.2; 3.1; 4.1\} \in \mathbf{M}$.

3. Schritt:

Berechne die Szenarioprobleme $(1.2), (2.2)$, übernehme $(3.1), (4.1)$ und erhalte die Lösung

$$
\begin{array}{llll}
 & & & y_3^1 = 0, \\
 & & x_2^1 = 0,\ x_2^2 = 0, & \\
 & & y_2^1 = 0,\ y_2^2 = 0, & \\
 & & & y_3^2 = 0, \\
x_1^1 = 0,\ x_1^2 = 0,\ x_1^3 = 0,\ x_1^4 = 1, & & & \\
 & & & y_3^3 = 0, \\
 & & x_2^3 = 1,\ x_2^4 = 0, & \\
 & & y_2^3 = 0,\ y_2^4 = 0, & \\
 & & & y_3^4 = 0,
\end{array}
$$

mit dem Zielfunktionswert $\varphi(\{1.2, 2.2, 3.1, 4.1\}) = -0.5$.

4. Schritt:
Alle Szenarioprobleme sind zulässig.

5. Schritt:
Inferiorität ist nicht gegeben.

6. Schritt:
Optimalität ist nicht gegeben.

7. Schritt:
Erhalte keine zulässige Lösung.

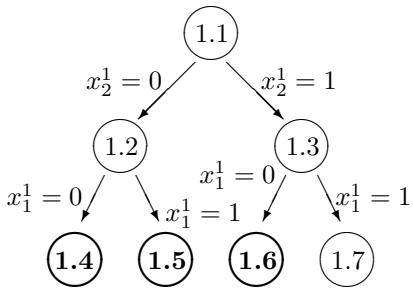
8.Schritt:
Ebene($\{1.3,2.3,3.1,4.1\}$)<MAXTIEFE $\rightarrow$ Fall (ii). Setze MAXTIEFE=2. Übernehme Familie $\mathcal{G}_1^1$ und Variable x_1 als Branchingentscheidung.

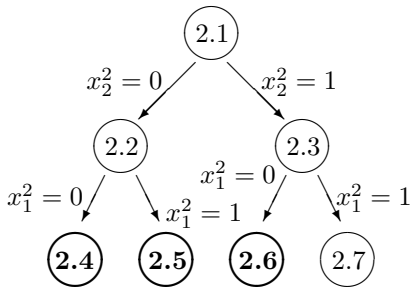
Iteration 5:
Es kam nun zum ersten Mal zur Situation, dass eine Branchingentscheidung nicht frei gewählt wurde, sondern eine vorgegebene Entscheidung aus der Branchingliste über-

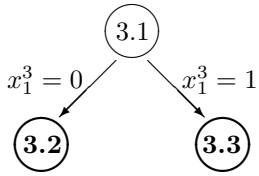
nommen werden musste. Bei der Erzeugung des Masterknotens $\{1.5, 2.5, 3.3, 4.3\}$ kam es zu Einspareffekten, da die Szenarioknoten (3.3) und (4.3) bereits berechnet wurden.

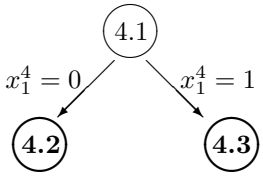
An dieser Stelle erkennt man deutlich, wieso eine einheitliche Branchingentscheidung auf jeder Ebene des Masterbaumes getroffen werden muss. Hätte man in der vorherigen Iteration 4 in Schritt 8 eine andere Branchingentscheidung gewählt, so hätten die Szenarioknoten (3.1) bzw. (4.1) andere Nachfolger bekommen als (3.2), (3.3) bzw. (4.2), (4.3). Somit wäre die binäre Baumstruktur der SBB's zerstört gewesen und es hätte innerhalb der SBB's eine Suche stattfinden müssen, da der rechte bzw. linke Nachfolger nicht mehr eindeutig gewesen wäre.

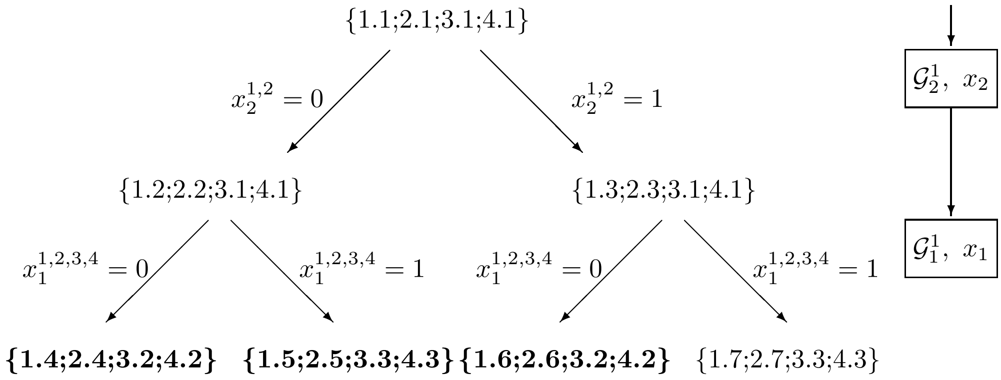
Außerdem sehen wir hier zum ersten Mal, dass Szenarioknoten wiederverwendet werden, die in einer vorherigen Iteration in keinem Masterknoten erhalten waren, vergleiche (3.3) und (4.3).











2. Schritt:

Wähle den Masterknoten $\{1.4; 2.4; 3.2; 4.2\} \in \mathbf{M}$.

3. Schritt:

Berechne die Szenarioprobleme $(1.4), (2.4), (3.2), (4.2)$ und erhalte die Lösung

$$y_3^1 = 0,$$
$$x_2^1 = 0,\ x_2^2 = 0,$$
$$y_2^1 = 0,\ y_2^2 = 0,$$
$$y_3^2 = 0,$$
$$x_1^1 = 0,\ x_1^2 = 0,\ x_1^3 = 0,\ x_1^4 = 0,$$
$$y_3^3 = 0,$$
$$x_2^3 = 0,\ x_2^4 = 1,$$
$$y_2^3 = 0,\ y_2^4 = 0,$$
$$y_3^4 = 0,$$

mit dem Zielfunktionswert $\varphi(\{1.4, 2.4, 3.2, 4.2\}) = -0.25$.

4. Schritt:

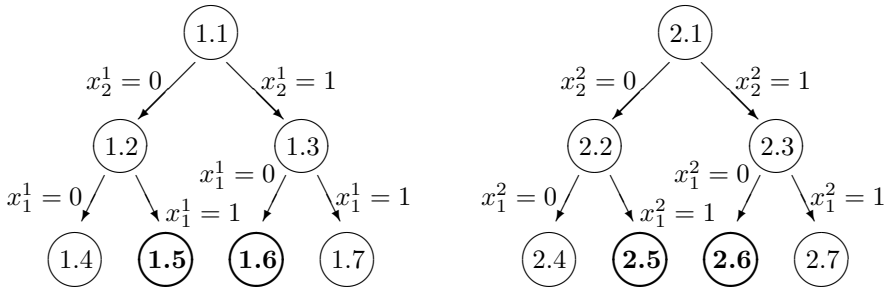
Alle Szenarioprobleme sind zulässig.

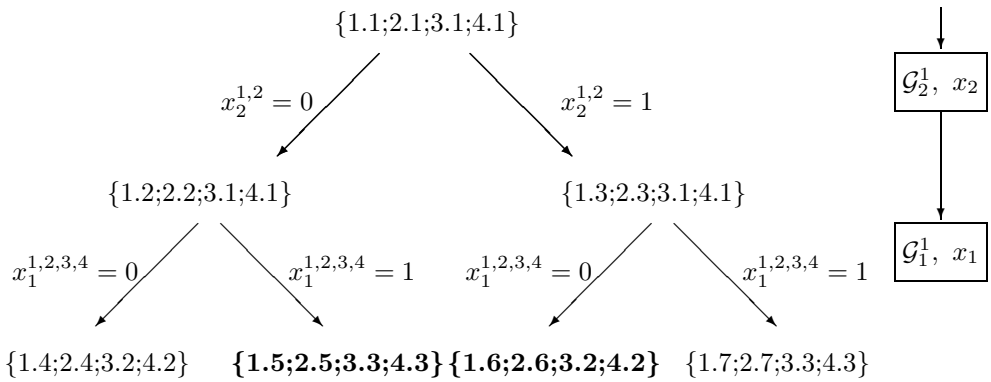
5. Schritt:

Inferiorität ist gegeben → gehe zu Schritt 2.

Iteration 6:

Es ergeben sich erneut Einspareffekte bei der Berechnung der Szenarioknoten, da die in der vorherigen Iteration berechneten Knoten (3.2) und (4.2) im Masterknoten $\{1.6, 2.6, 3.2, 4.2\}$ verwendet werden können.





2. Schritt:
Wähle den Masterknoten $\{1.5; 2.5; 3.3; 4.3\} \in \mathbf{M}$.

3. Schritt:
Berechne die Szenarioprobleme $(1.5), (2.5)$ übernehme $, (3.3), (4.3)$ und erhalte die Lösung

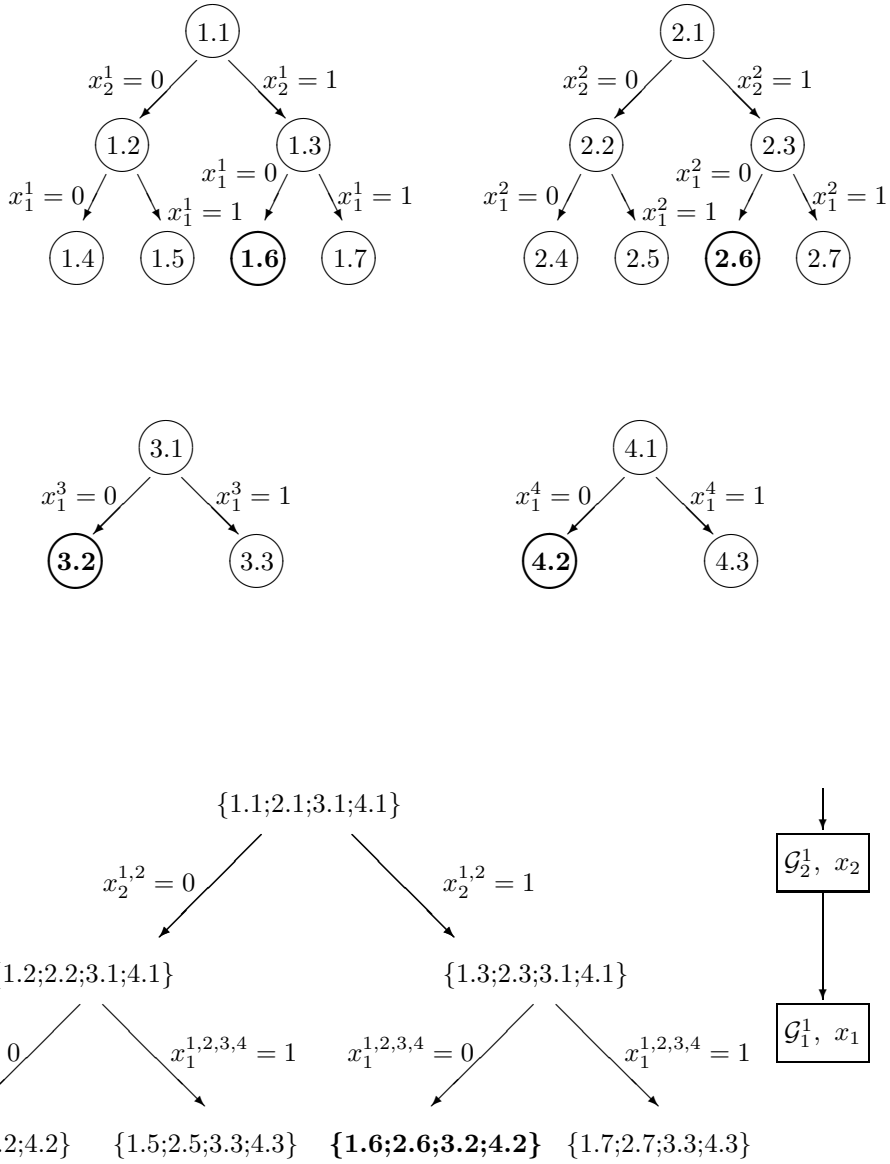
$$\begin{array}{lll} & & y_3^1 = 1, \\ & x_2^1 = 0,\ x_2^2 = 0, & \\ & y_2^1 = 1,\ y_2^2 = 1, & \\ & & y_3^2 = 0, \\ x_1^1 = 1,\ x_1^2 = 1,\ x_1^3 = 1,\ x_1^4 = 1, & & \\ & & y_3^3 = 1, \\ & x_2^3 = 0,\ x_2^4 = 0, & \\ & y_2^3 = 0,\ y_2^4 = 0, & \\ & & y_3^4 = 0, \end{array}$$

mit dem Zielfunktionswert $\varphi(\{1.5, 2.5, 3.3, 4.3\}) = 2.25$.

4. Schritt:
Alle Szenarioprobleme sind zulässig.

5. Schritt:
Inferiorität ist gegeben $\rightarrow$ gehe zu Schritt 2.

Iteration 7:
Es wurden der Masterbaum und die Szenariobäume entsprechend aufdatiert. Die Szenarioknoten (3.3) und (4.3) können nun abgeschnitten werden.



2. Schritt:

Wähle den Masterknoten $\{1.6; 2.6; 3.2; 4.2\} \in \mathbf{M}$.

3. Schritt:

Berechne die Szenarioprobleme $(1.6), (2.6)$ übernehme $, (3.2), (4.2)$ und erhalte die Lösung

$$
\begin{array}{lll}
 & & y_3^1 = 1, \\
 & x_2^1 = 1,\ x_2^2 = 1, & \\
 & y_2^1 = 0,\ y_2^2 = 0, & \\
 & & y_3^2 = 0, \\
x_1^1 = 0,\ x_1^2 = 0,\ x_1^3 = 0,\ x_1^4 = 0, & & \\
 & & y_3^3 = 0, \\
 & x_2^3 = 0,\ x_2^4 = 1, & \\
 & y_2^3 = 0,\ y_2^4 = 0, & \\
 & & y_3^4 = 0,
\end{array}
$$

mit dem Zielfunktionswert $\varphi(\{1.6, 2.6, 3.2, 4.2\}) = -0.375$.

4. Schritt:
Alle Szenarioprobleme sind zulässig.

5. Schritt:
Inferiorität ist nicht gegeben.

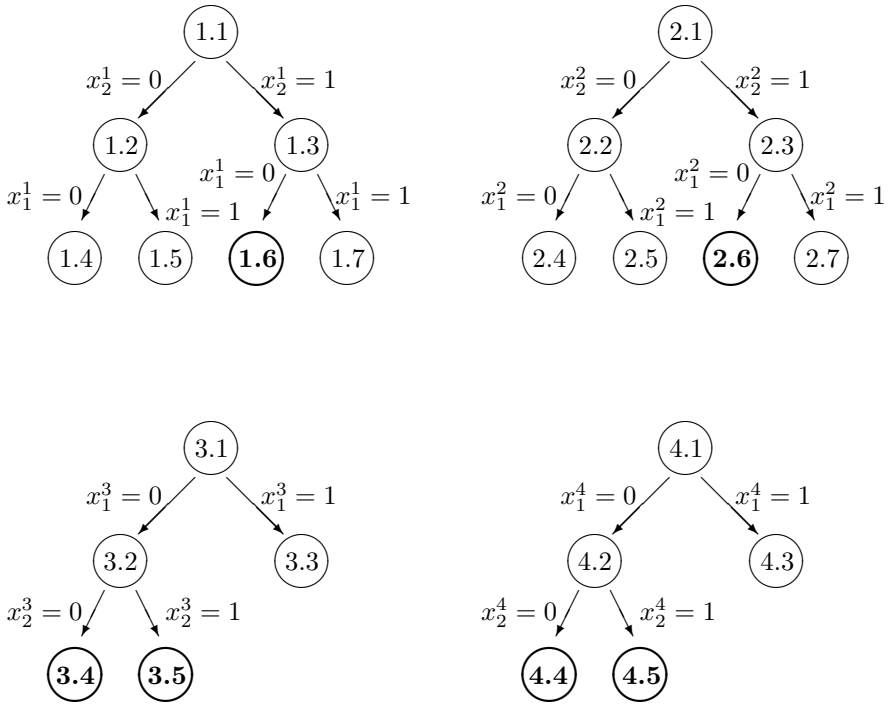
6. Schritt:
Optimalität ist nicht gegeben

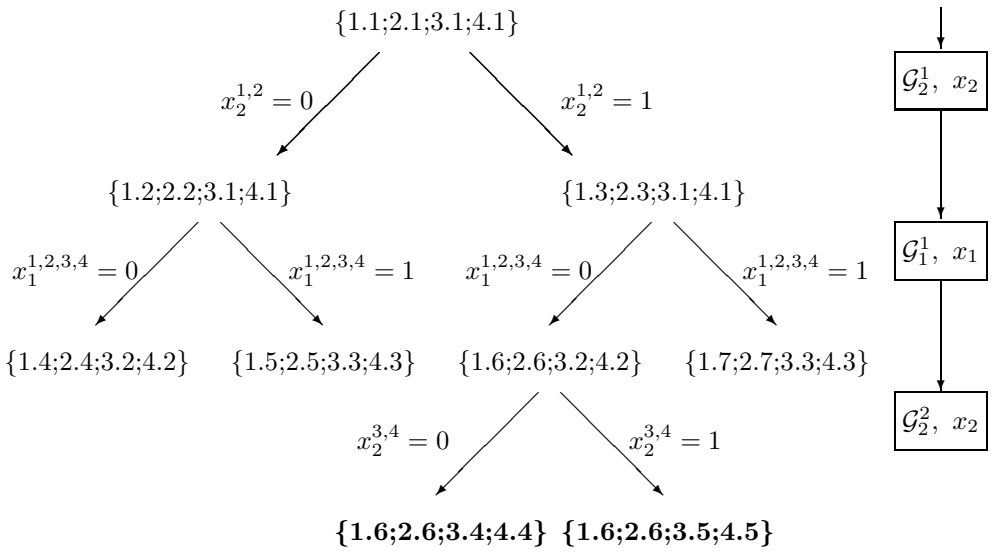
7. Schritt:
Erhalte keine zulässige Lösung.

8.Schritt:
Ebene({1.6,2.6,3.2,4.2})=MAXTIEFE → Fall (i). Setze MAXTIEFE=3. Wähle Familie $\mathcal{G}_2^2$ und Variable x_2.

Iteration 8:
Da ein Masterknoten auf einer neuen Ebene erzeugt wurde, wurde eine neue Bran-

chingentscheidung getroffen und in der Branchingliste gespeichert. Wiederum kommt es zu Einspareffekten bei den Szenarioknoten, da in den neu erzeugten Masterknoten die Szenarioknoten (1.6) und (2.6) bereits berechnet wurden.





2. Schritt:

Wähle den Masterknoten $\{1.6; 2.6; 3.4; 4.4\} \in \mathbf{M}$.

3. Schritt:

Berechne die Szenarioprobleme $(3.4), (4.4)$ übernehme $, (1.6), (2.6)$ und erhalte die Lösung

$$y_3^1 = 1,$$
$$x_2^1 = 1,\ x_2^2 = 1,$$
$$y_2^1 = 0,\ y_2^2 = 0,$$
$$y_3^2 = 0,$$
$$x_1^1 = 0,\ x_1^2 = 0,\ x_1^3 = 0,\ x_1^4 = 0,$$
$$y_3^3 = 0,$$
$$x_2^3 = 0,\ x_2^4 = 0,$$
$$y_2^3 = 0,\ y_2^4 = 0,$$
$$y_3^4 = 0,$$

mit dem Zielfunktionswert $\varphi(\{1.5, 2.5, 3.3, 4.3\}) = -0.125$.

4. Schritt:

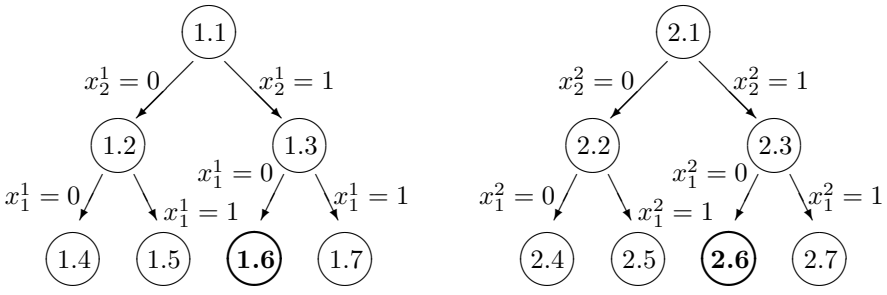
Alle Szenarioprobleme sind zulässig.

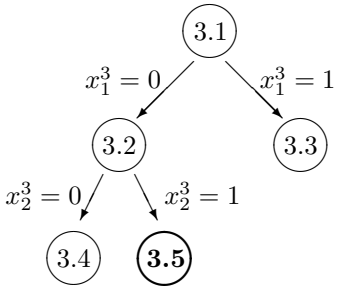
5. Schritt:

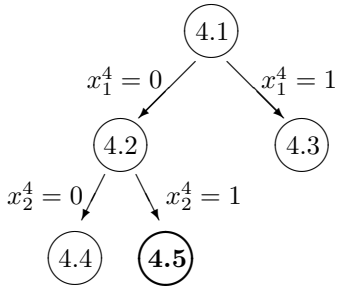
Inferiorität ist gegeben $\rightarrow$ gehe zu Schritt 2.

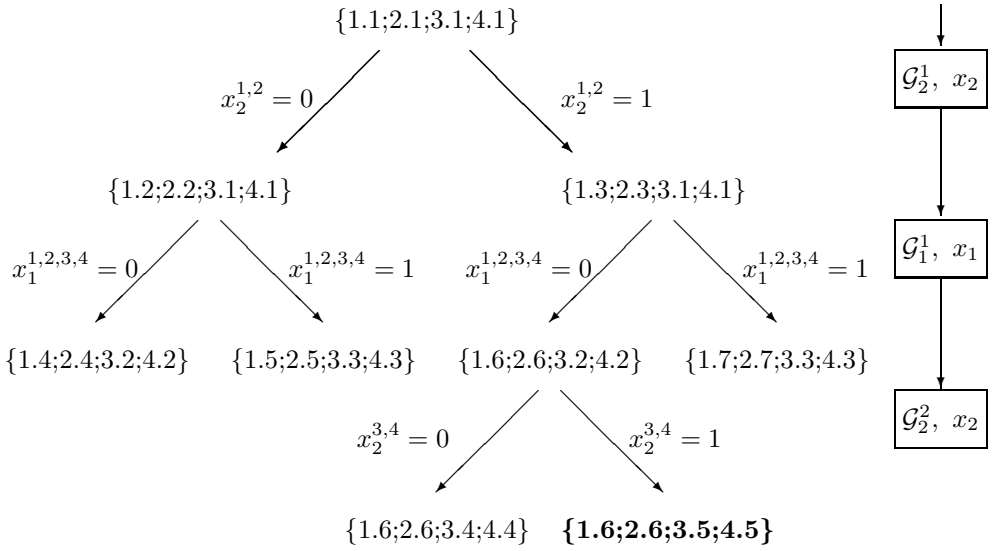
Iteration 9:

Der Masterknoten und die nicht mehr vorkommenden Szenarioknoten wurden gelöscht.









2. Schritt:

Wähle den Masterknoten $\{1.6; 2.6; 3.5; 4.5\} \in \mathbf{M}$.

3. Schritt:

Berechne die Szenarioprobleme $(3.5), (4.5)$ übernehme , $(1.6), (2.6)$ und erhalte die Lösung

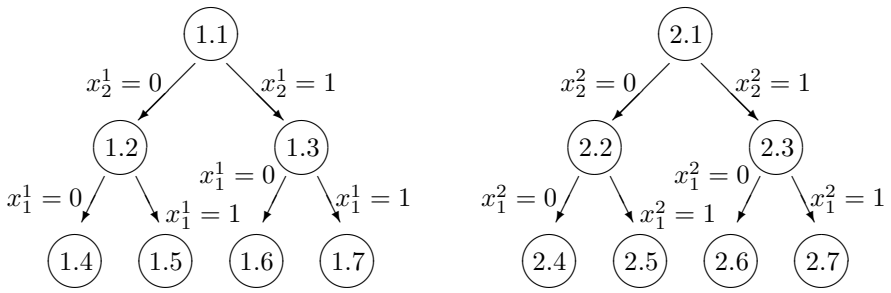
$$y_3^1 = 1,$$
$$x_2^1 = 1,\ x_2^2 = 1,$$
$$y_2^1 = 0,\ y_2^2 = 0,$$
$$y_3^2 = 0,$$
$$x_1^1 = 0,\ x_1^2 = 0,\ x_1^3 = 0,\ x_1^4 = 0,$$
$$y_3^3 = 1,$$
$$x_2^3 = 1,\ x_2^4 = 1,$$
$$y_2^3 = 0,\ y_2^4 = 0,$$
$$y_3^4 = 0,$$

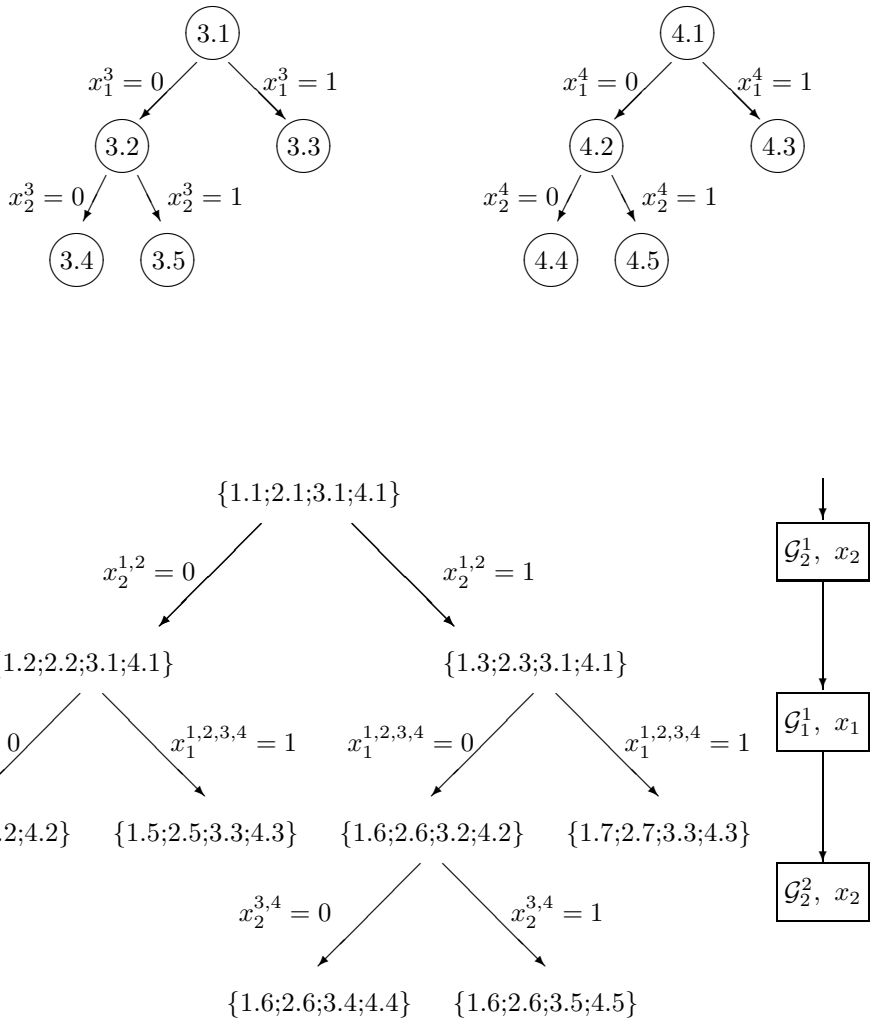
mit dem Zielfunktionswert $\varphi(\{1.5, 2.5, 3.3, 4.3\}) = -0.25$.

4. Schritt:
Alle Szenarioprobleme sind zulässig.

5. Schritt:
Inferiorität ist gegeben → gehe zu Schritt 2.

Iteration 10:
Mit dem letzten Masterknoten wurden auch alle Szenarioknoten gelöscht.





2. Schritt:
$\mathbf{M} = \emptyset \rightarrow$ STOPP. $z = -0.25$ ist optimal.

Ingesamt wurden in diesem kleinen Beispiel 33% weniger Szenarioknoten berechnet als in der der Basisversion des Algorithmus in Kapitel 3.3. Das heißt, die Anzahl an zu berechnenden Szenarioknoten ohne Verwaltungssystem (#Masterknoten $\times$ #Szenarien) entspricht 100%. Bei großen Problemen nimmt dieser Effekt drastisch zu, wie in Kapitel 4 noch berichtet wird.

3.3.2 bb-hash

Eine Alternative zum obigen Konzept, bietet das Speichern der Szenarioknoten in einer Liste (ohne Struktur). Wie die einzelnen Knoten gespeichert werden, hängt dann von dem Speicherverfahren ab. In dieser Arbeit wurde das Streuspeicherverfahren (engl. hashing) benutzt, siehe etwa [19, 35]. Bei diesem Verfahren wird eine Tabelle H mit A Adressen vordefiniert. Die Adressen der einzelnen Szenarioknoten SK werden hierbei mit einer Hash-Funktion oder Speicherfunktion

$$h : SK \longrightarrow \{1, \ldots, A\} \tag{3.10}$$

kodiert. Die Suche nach geeigneten Hash-Funktionen ist ein Forschungsgebiet innerhalb der kombinatorischen Optimierung, siehe etwa [25, 47]. Die Kodierung erfolgt mithilfe der Daten der Szenarioknoten. Die Informationen, die ein Szenarioknoten in unserer Implementierung enthält, zeigt folgende Abbildung:

Szenario	Zähler	Länge	Index 1	GK 1	Wert 1
			Index 2	GK 2	Wert 2
			Index 3	GK 3	Wert 3
			Index 4	GK 4	Wert 4

Abb. 4.10: Datensatz

Im Feld 'Szenario' steht das Szenario des Szenarioknotens. Im Feld 'Zähler' ist angegeben in wievielen Masterknoten dieser Szenarioknoten aktuell enthalten ist. Der Wert 'Länge' im nächsten Feld gibt die Anzahl der gespeicherten Branchingentscheidungen an. In diesem Beispiel sind es vier. Im Allgemeinen ist die Anzahl beliebig. In den nächsten Feldern sind die Informationen über die Branchingentscheidungen gespeichert. Im ersten Feld 'Index 1', der Index der ersten Branchingentscheidung, im Feld 'GK 1' das größer oder kleiner Zeichen, im Feld 'Wert 1' die Schranke. Entsprechendes gilt für die nächsten drei gespeicherten Entscheidungen.
Aus diesem Datensatz wurden die folgenden Hash-Funktionen implementiert:

$$h_1(SK) := (a_1\text{Szenario} + a_2\text{Länge})_{mod\ A} \ \text{ mit } \ a_1, a_2 \in \mathbb{N} \tag{3.11}$$

$$h_2(SK) := (\text{Szenario}^{\text{Länge}})_{mod\ A}. \tag{3.12}$$

Für ein $n \in \mathbb{N}$ ist hier $(n)_{mod\ A}$ die Modulo-Berechnung von n bezüglich A. Diese garantiert, dass $h(SK) \in \{1, \ldots, A\}$, für alle zu betrachtenden Szenarioknoten.
Eine Kollision ist gegeben, wenn zwei verschiedene Szenarioknoten auf dieselbe Adresse abgebildet werden. Da die Größe der Tabelle H zu Beginn des Programmstarts festgelegt werden muss und die Anzahl der erzeugten Szenarioknoten zu diesem Zeitpunkt nicht bekannt sind, sind Kollisionen eher die Regel als die Ausnahme. Kollisionen werden wie folgt behandelt: Mithilfe des Datensatzes der Szenarioknoten wird eine lexikographische Ordnung definiert, die es erlaubt, zwei unterschiedlichen Szenarioknoten eine Relationen '<' bzw. '>' zuzuordnen. Folgende lexikographische Prioritätsordnung wurde gewählt:

1.) 'Szenario',

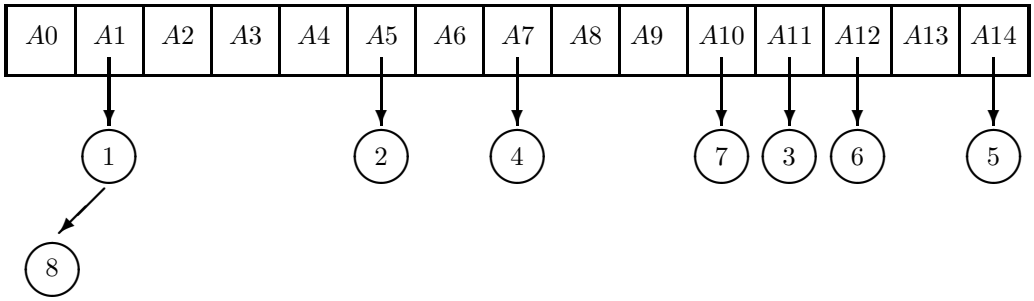
2.) 'Länge',

3.) 'Wert 1','Wert 2',...,'Wert Länge',

4.) 'Index 1','Index 2',...,'Index Länge',

5.) 'GK 1','GK 2',...,'GK Länge',

wobei bei den ersten vier Punkten ein Zahlenvergleich stattfindet, bei Punkt 5.) die Ordnung $'' \leq'' < '' \geq''$ gilt. Bei einer Kollision wird an der entsprechenden Kollisionsadresse ein binärer Baum konstruiert, bei dem ein 'kleinerer' Szenarioknoten zum linken- und ein 'größerer' Szenarioknoten zum rechten Nachfolger des bereits gespeicherten Szenarioknotens wird. Folgende Graphik verdeutlicht dieses:



Zu sehen ist hier eine Tabelle mit A=15 Adressen. Dem Szenarioknoten 8 wurde wie dem Szenarioknoten 1 mit der Hash-Funktion die Adresse A1 zugeordnet. Der lexikographische Vergleich ergab: Szenarioknoten 8 < Szenarioknoten 1. Deshalb wurde Szenarioknoten 8 zum linken Nachfolger des Szenarioknotens 1. Bei der nächsten Kollision in A1 wird der neue Szenarioknoten mit Szenarioknoten 1 verglichen. Ist dieser größer, so wird er zum rechten Nachfolger von Szenarioknoten 1. Ist dieser kleiner findet ein weiterer Vergleich mit Szenarioknoten 8 statt usw. Durch diese Anordnung wird ein schnelleres Suchen innerhalb einer Liste ermöglicht.

Der Vorteil der Speicherverwaltung liegt in der relativen Unabhängigkeit der Suche zur Anzahl der Daten. Der entscheidende Unterschied zu bb-tree ist, dass durch die unstruktuierte Speicherverwaltung -keine binären SBB's wie in bb-tree- die Branchingentscheidungen in Schritt 8 unabhängig von den vorherigen Entscheidungen gewählt werden können. Allerdings nimmt man den Nachteil in Kauf, dass die Szenarioknoten erst gesucht werden müssen, und nicht durch die Struktur der Speicherung abrufbar sind.

Da es beim Abschneiden der Szenarioknoten keine effiziente, nachweisbar korrekte Regel gibt wie bei 'bb-tree' - abschneiden des Szenarioknotens, falls er in keinem Masterknoten vorhanden ist und der Vorgängerknoten bereits abgeschnitten wurde - ist folgendes Verfahren implementiert worden:
Nach einer gewissen Anzahl an Iterationen, die vor dem Start festgelegt wird, schneide alle Szenarienknoten ab, die in keinem aktiven Masterknoten enthalten sind.

Dabei kann es durchaus vorkommen, dass Szenarioknoten gelöscht werden, die in einer späteren Iteration des Algorithmus erneut gebraucht werden. Vergleiche hierzu die Iterationen 4 und 5 im Beispiel des vorangegangenen Abschnittes. In Iteration 4 waren die Szenarioknoten (3.3) und (4.3) in keinem Masterknoten enthalten. Hätte man dieses Verfahren in dieser Iteration ausgeführt, hätten diese Knoten in Iteration 5 erneut berechnen werden müssen.

'bb-hash'

SCHRITT 1: (Initialisierung):
Setze Masterproblem M auf Masterliste **M** und Szenarioprobleme in die Hash-Tabelle **H**. Setze z:=$+\infty$.

SCHRITT 2: (Knotenauswahl):
Falls $\mathbf{M} = \emptyset$, TERMINATE. Falls nicht, wähle nächstes Masterproblem $M \in \mathbf{M}$ und die zugehörigen Szenarioprobleme $P_M^1, \ldots, P_M^S \in \mathbf{H}$. Lösche M aus Liste.

SCHRITT 3: (Berechnung):
Berechne alle nicht bereits berechneten Szenarioprobleme.

SCHRITT 4: (Zulässigkeit):
Falls gewisse P_M^s unzulässig sind, lösche diese Probleme aus der Hash-Tabelle, lösche alle $M' \in \mathbf{M}$, welche die entsprechenden Szenarioprobleme enthalten und gehe zu SCHRITT2. Andernfalls berechne den Zielfunktionswert $\varphi(M)$ zur Lösung $x(M)$.

SCHRITT 5: (Inferiorität):
Falls z $\leq \varphi(M)$, gehe zu SCHRITT 2.

SCHRITT 6: (Optimalität):
Falls $x(M)$ die relaxierten Bedingungen erfüllt, setze z = $\varphi(M)$, lösche alle $M' \in \mathbf{M}$ mit LB(M') $\geq \varphi(M)$, gehe zu SCHRITT 2.

SCHRITT 7: (Heuristik):
Versuche mittels Heuristiken, aus $x(M)$ eine zulässige Lösung mit Zielfunktionswert $h(M)$ zu generieren. Falls $h(M) < z$, setze $z = h(M)$, lösche alle $M' \in \mathbf{M}$ mit LB(M') $\geq h(M)$.

SCHRITT 8: (Branching):
Treffe eine Branchingentscheidung. Es sei $\mathcal{G}_t^r$ die zu branchende Familie mit dem Index (i). Erzeuge für alle $P_M^s,\ s \in \mathcal{G}_t^r$ zwei neue Probleme $P1^s, P2^s, s \in \mathcal{G}_t^r$ mit den zusätzlichen Bedingungen:

$$x_{t,(i)}^s \leq \lfloor x_B \rfloor \text{ falls } x_{t,(i)}^s \text{ integer},\ x_{t,(i)}^s \leq x_B - \gamma \text{ sonst},\ s \in \mathcal{G}_t^r$$

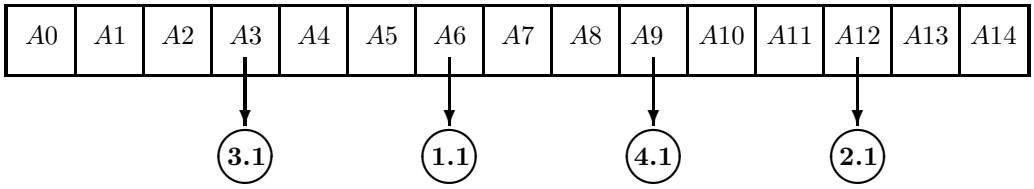
bzw

$$x_{t,(i)}^s \geq \lceil x_B \rceil \text{ falls } x_{t,(i)}^s \text{ integer},\ x_{t,(i)}^s \leq x_B + \gamma \text{ sonst},\ s \in \mathcal{G}_t^r$$

für ein gegebenes $\gamma > 0$. Kontrolliere, ob die erzeugten Szenarioknoten in **H** bereits existieren, falls nicht, ordne sie in der Hash-Tabelle ein. Erzeuge die entsprechenden Masterknoten im Masterbaum und gehe zu SCHRITT 2.

Auch dieser Algorithmus soll mithilfe von Beispiel 3.4 demonstriert werden.

Iteration 1:
Wir betrachten ein Tabelle mit A=15 Adressen. Die Adressen der Szenarioprobleme wurden mit der Hash-Funktion 3.11 und den Gewichten $a_1 = 6$ und $a_2 = 2$ berechnet und entsprechend in der Tabelle einsortiert. Wie in obiger Notation ist hier der Wurzelknoten des Masterbaumes dargestellt. Eine Branchingliste wie bei bb-tree ist nicht notwendig.



{1.1;2.1;3.1;4.1}

1. Schritt:

Setze $z = -\infty$,

$\mathbf{M} = \{\{1.1, 2.1, 3.1, 4.1\}\}$,

$\mathbf{H} = \{(1.1), (2.1), (3.1), (4.1)\}$.

2. Schritt:

Wähle den Masterknoten $\{1.1, 2.1, 3.1, 4.1\} \in \mathbf{M}$.

3. Schritt:

Berechne die Szenarioprobleme $(1.1), (2.1), (3.1), (4.1)$ und erhalte die Lösung

$$y_3^1 = 0,$$
$$x_2^1 = 0,\ x_2^2 = 1,$$
$$y_2^1 = 0,\ y_2^2 = 0,$$
$$y_3^2 = 0,$$
$$x_1^1 = 0,\ x_1^2 = 0,\ x_1^3 = 0,\ x_1^4 = 1,$$
$$y_3^3 = 0,$$
$$x_2^3 = 0,\ x_2^4 = 0,$$
$$y_2^3 = 0,\ y_2^4 = 0,$$
$$y_3^4 = 0,$$

mit dem Zielfunktionswert $\varphi(\{1.1, 2.1, 3.1, 4.1\}) = -0.5$.

4. Schritt:
Alle Szenarioprobleme sind zulässig.

5. Schritt:
Inferiorität ist nicht gegeben.

6. Schritt:
Optimalität ist nicht gegeben.

7. Schritt:
Erhalte aus der Heuristik die zulässige Lösung

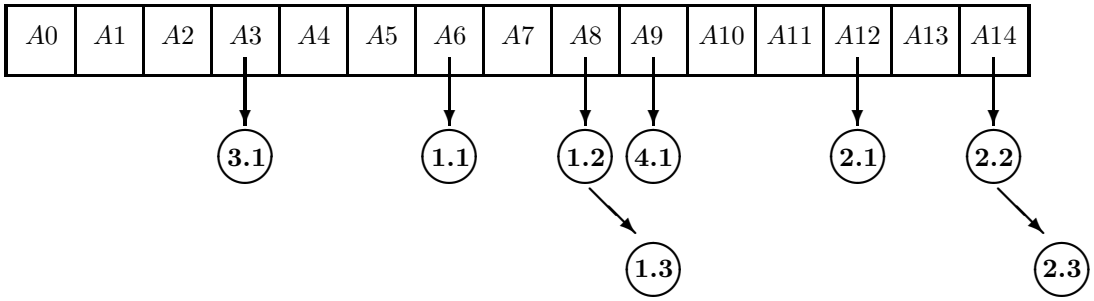
$$\begin{array}{rrrr} & & & y_3^1 = 1, \\ & & x_2^1 = 1,\ x_2^2 = 1, & \\ & & y_2^1 = 0,\ y_2^2 = 0, & \\ & & & y_3^2 = 0, \\ x_1^1 = 0,\ x_1^2 = 0,\ x_1^3 = 0,\ x_1^4 = 0, & & & \\ & & & y_3^3 = 1, \\ & & x_2^3 = 1,\ x_2^4 = 1, & \\ & & y_2^3 = 0,\ y_2^4 = 0, & \\ & & & y_3^4 = 0, \end{array}$$

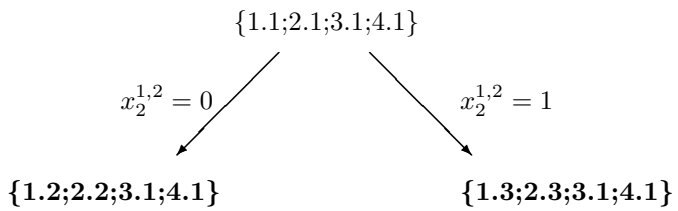
mit dem Zielfunktionswert h({1.1,2.1,3.1,4.1})=-0.25. Setze z=-0.25.

8.Schritt:
Wähle Familie $\mathcal{G}_2^1$ und Variable x_2. Erzeuge die Szenarioknoten $(1.2), (1.3)$ und $(2.1), (2.3)$ und überprüfe, ob diese bereits in $\mathbf{H}$ enthalten sind. Falls nicht, berechne die Adressen und ordne sie entsprechend ein.

Iteration 2:

Anders als bei bb-tree wird die Branchingentscheidung nicht gespeichert. Alle Szenarioknoten sind fett markiert, da sie im Weiteren Verlauf des Algorithmus wiederverwendet werden können; ein Löschen erfolgt nur - wenn überhaupt - in bestimmten Iterationen, die fest vorgegeben sind. Die Hash-Adressen der Szenarionknoten (1.2), (1.3) und (2.2), (2.3) sind identisch. Es kommt also zu Kollisionen. Der Vergleich der Knoten ergibt (1.2) $<$ (1.3) und (2.2) $<$ (2.3). Die Knoten (1.2) und (2.2) werden zuerst in die Tabelle einsortiert. Somit sind (1.3) bzw. (2.3) die rechten Nachfolger. Im Masterbaum wurde der Wurzelknoten gemäß des üblichen Branch-and-Bound Schemas gelöscht. Wie bei bb-tree erzielt man dieselben Einspareffekte, denn zum erzeugen der Masterknoten $\{1.2, 2.2, 3.1, 4.1\}$ und $\{1.3, 2.3, 3.1, 4.1\}$ können die bereits berechneten Szenarioknoten ebenfalls (3.1) und (4.1) wieder verwendet werden.





2. Schritt:

Wähle den Masterknoten $\{1.3, 2.3, 3.1, 4.1\} \in \mathbf{M}$.

3. Schritt:
Berechne die Szenarioprobleme $(1.3), (2.3)$, übernehme $(3.1), (4.1)$ und erhalte die Lösung

$$\begin{array}{llll} & & & y_3^1 = 1, \\ & & x_2^1 = 1,\ x_2^2 = 1, & \\ & & y_2^1 = 0,\ y_2^2 = 0, & \\ & & & y_3^2 = 0, \\ x_1^1 = 0,\ x_1^2 = 0,\ x_1^3 = 0,\ x_1^4 = 1, & & & \\ & & & y_3^3 = 0, \\ & & x_2^3 = 0,\ x_2^4 = 0, & \\ & & y_2^3 = 0,\ y_2^4 = 0, & \\ & & & y_3^4 = 0, \end{array}$$

mit dem Zielfunktionswert $\varphi(\{1.3, 2.3, 3.1, 4.1\}) = -0.375$.

4. Schritt:
Alle Szenarioprobleme sind zulässig.

5. Schritt:
Inferiorität ist nicht gegeben

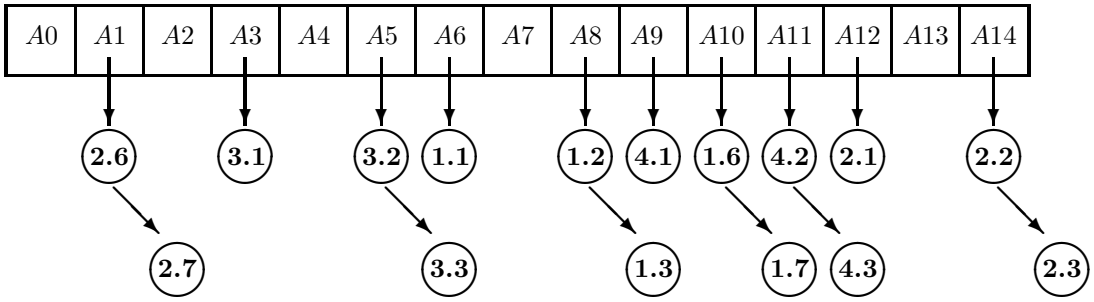
6. Schritt:
Optimalität ist nicht gegeben

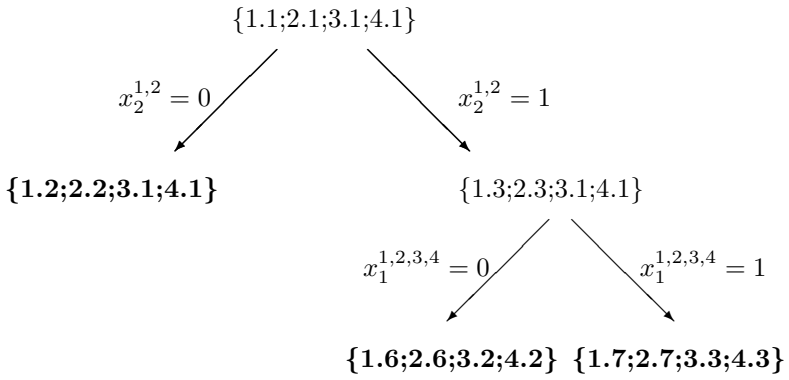
7. Schritt:
Erhalte keine zulässige Lösung.

8.Schritt:
Wähle Familie $\mathcal{G}_1^1$ und Variable x_1.
Erzeuge die Szenarioknoten (1.6),(1.7),(2.6), (2.7),(3.2),(3.3),(4.2),(4.3) und überprüfe ob diese bereits in $\mathbf{H}$ enthalten sind. Falls nicht, berechne die Adressen und ordne sie entsprechend ein.

Iteration 3:

Wieder gab es eine Kollision bei den Adressen $A1$, $A5$, $A10$ und $A11$, diese wurde gemäß der Lexikographie behandelt. Zu Einspareffekten kam es in dieser Iteration nicht.





2. Schritt:

Wähle den Masterknoten $\{1.7; 2.7; 3.3; 4.3\} \in \mathbf{M}$.

3. Schritt:

Berechne die Szenarioprobleme $(1.7), (2.7), (3.3), (4.3)$ und erhalte die Lösung

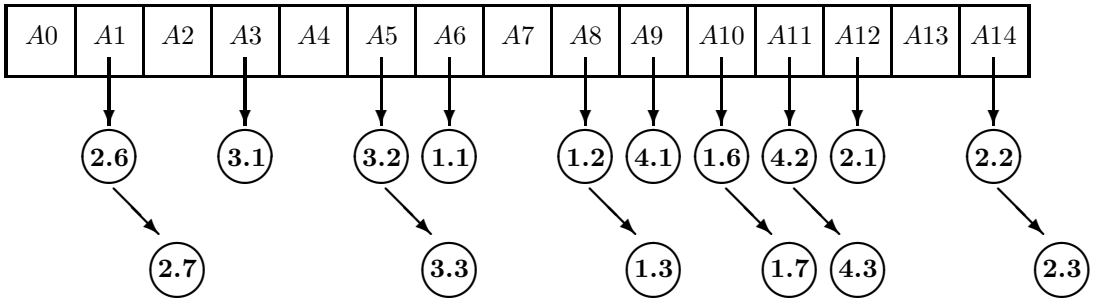
$$y_3^1 = 1,$$
$$x_2^1 = 0,\ x_2^2 = 0,$$
$$y_2^1 = 1,\ y_2^2 = 1,$$
$$y_3^2 = 0,$$
$$x_1^1 = 1,\ x_1^2 = 1,\ x_1^3 = 1,\ x_1^4 = 1,$$
$$y_3^3 = 1,$$
$$x_2^3 = 1,\ x_2^4 = 1,$$
$$y_2^3 = 0,\ y_2^4 = 0,$$
$$y_3^4 = 0,$$

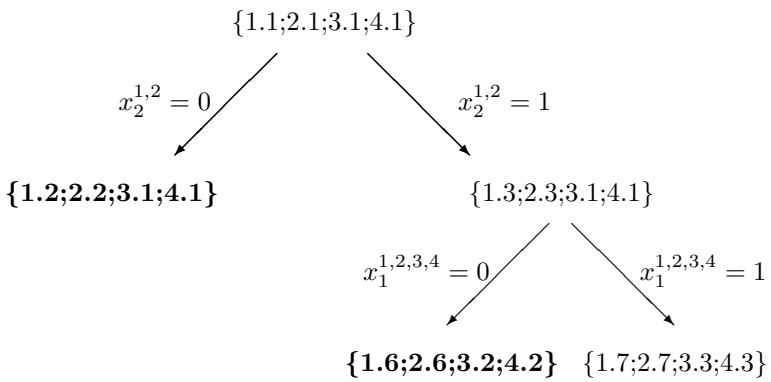
mit dem Zielfunktionswert $\varphi(\{1.7, 2.7, 3.3, 4.3\}) = 2.5$

4. Schritt:
Alle Szenarioprobleme sind zulässig.

5. Schritt:
Inferiorität ist gegeben → gehe zu Schritt 2.

Iteration 4:
Der unterlegende Masterknoten wurde entsprechend des Branch-and-Bound Schemas gelöscht.





2. Schritt:
Wähle den Masterknoten $\{1.2; 2.2; 3.1; 4.1\} \in \mathbf{M}$.

3. Schritt:
Berechne die Szenarioprobleme $(1.2), (2.2)$, übernehme $(3.1), (4.1)$ und erhalte die Lösung

$$\begin{array}{rl} & y_3^1 = 0, \\ x_2^1 = 0,\ x_2^2 = 0, & \\ y_2^1 = 0,\ y_2^2 = 0, & \\ & y_3^2 = 0, \\ x_1^1 = 0,\ x_1^2 = 0,\ x_1^3 = 0,\ x_1^4 = 1, & \\ & y_3^3 = 0, \\ x_2^3 = 1,\ x_2^4 = 0, & \\ y_2^3 = 0,\ y_2^4 = 0, & \\ & y_3^4 = 0, \end{array}$$

mit dem Zielfunktionswert $\varphi(\{1.2, 2.2, 3.1, 4.1\}) = -0.5$.

4. Schritt:
Alle Szenarioprobleme sind zulässig.

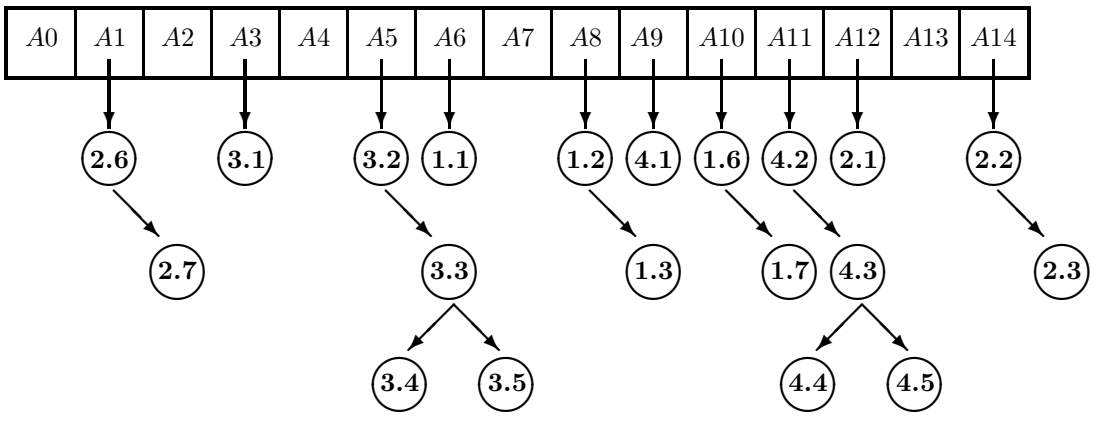
5. Schritt:
Inferiorität ist nicht gegeben.

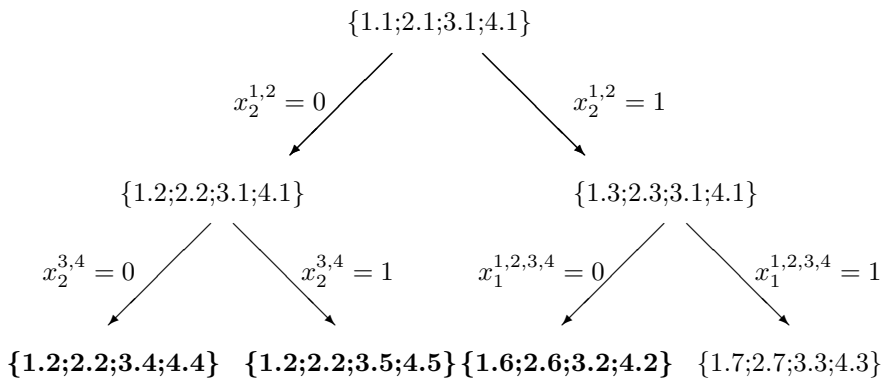
6. Schritt:
Optimalität ist nicht gegeben.

7. Schritt:
Erhalte keine zulässige Lösung.

8.Schritt:
Wähle Familie $\mathcal{G}_2^2$ und Variable x_2.
Erzeuge die Szenarioknoten (3.4),(3.5) und (4.4), (4.5), und überprüfe ob diese bereits in **H** enthalten sind. Falls nicht, berechne die Adressen und ordne sie entsprechend ein.

Iteration 5:
Wir haben hier nun die Situation, dass an zwei Masterknoten auf einer Ebene verschiedene Branchingstrategien ausgeübt werden, was bei bb-tree nicht möglich ist. Außerdem kam es zu weiteren Einspareffekten, da die Knoten (1.2), (2.2) bereits berechnet wurden.





2. Schritt:
Wähle den Masterknoten $\{1.2; 2.2; 3.4; 4.4\} \in \mathbf{M}$.

3. Schritt:
Berechne die Szenarioprobleme $(3, 4), (4.4)$, übernehme $(1, 2), (2.2)$ und erhalte die Lösung

$$\begin{array}{rl} & y_3^1 = 0, \\ x_2^1 = 0,\ x_2^2 = 0, & \\ y_2^1 = 0,\ y_2^2 = 0, & \\ & y_3^2 = 0, \\ x_1^1 = 0,\ x_1^2 = 0,\ x_1^3 = 0,\ x_1^4 = 1, & \\ & y_3^3 = 0, \\ x_2^3 = 0,\ x_2^4 = 0, & \\ y_2^3 = 0,\ y_2^4 = 0, & \\ & y_3^4 = 0, \end{array}$$

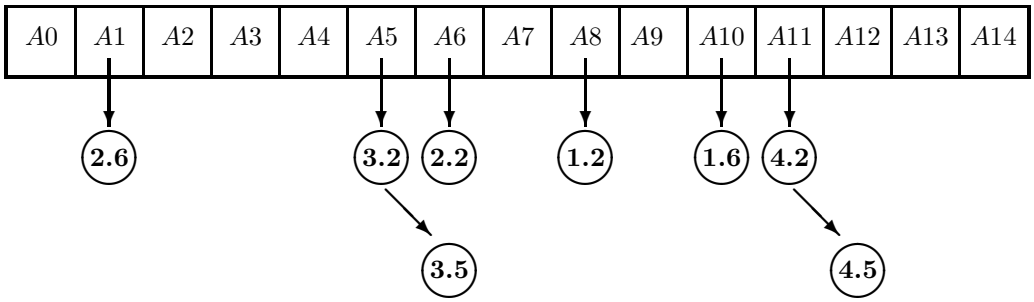
mit dem Zielfunktionswert $\varphi(\{1.2, 2.2, 3.4, 4.4\}) = -0.25$.

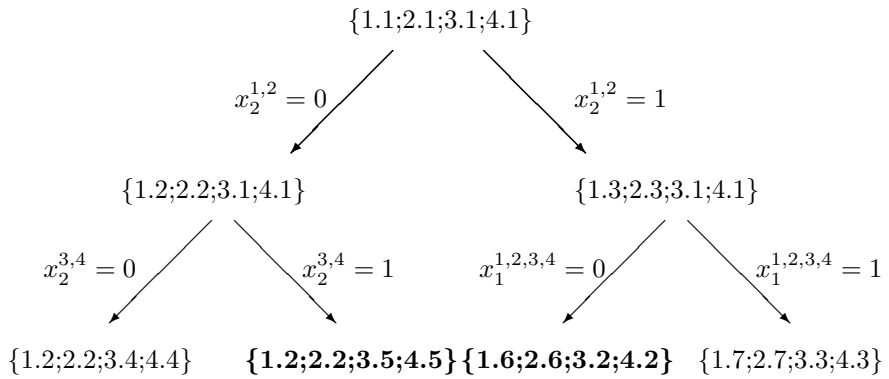
4. Schritt:

Alle Szenarioprobleme sind zulässig.

5. Schritt:
Inferiorität ist gegeben → gehe zu Schritt 2.

Iteration 6:
In dieser Iteration werden alle Szenarioknoten, die in keinem Masterknoten vorhanden sind gelöscht. Wie bereits beschrieben, ist diese Entscheidung rein optional. Bei den meisten Testrechnungen hat es sich gezeigt, dass ein Abschneiden nicht nötig war. Ab einer bestimmten Tiefe des Masterbaumes wurden so gut wie keine Szenarioknoten mehr berechnet.





2. Schritt:

Wähle den Masterknoten $\{1.2; 2.2; 3.5; 4.5\} \in \mathbf{M}$.

3. Schritt:

Berechne die Szenarioprobleme $(3.5), (4.5)$ übernehme $, (1.2), (2.2)$ und erhalte die Lösung

$$y_3^1 = 0,$$
$$x_2^1 = 0,\ x_2^2 = 0,$$
$$y_2^1 = 0,\ y_2^2 = 0,$$
$$y_3^2 = 0,$$
$$x_1^1 = 0,\ x_1^2 = 0,\ x_1^3 = 0,\ x_1^4 = 0,$$
$$y_3^3 = 1,$$
$$x_2^3 = 1,\ x_2^4 = 1,$$
$$y_2^3 = 0,\ y_2^4 = 0,$$
$$y_3^4 = 0,$$

mit dem Zielfunktionswert $\varphi(\{1.2, 2.2, 3.5, 4.5\}) = -0.125$.

4. Schritt:

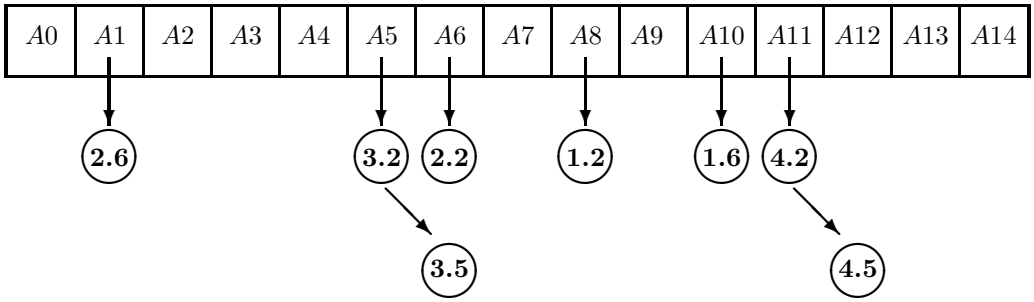
Alle Szenarioprobleme sind zulässig.

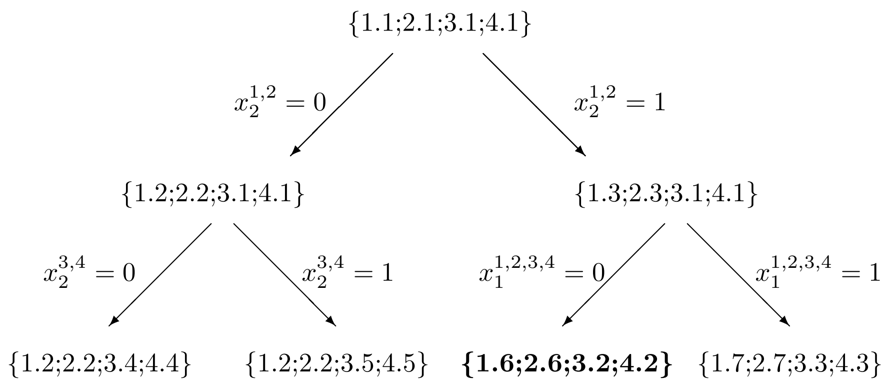
5. Schritt:

Inferiorität ist gegeben $\rightarrow$ gehe zu Schritt 2.

Iteration 7:

Es wurden der Masterbaum und die Szenariobäume entsprechend aufdatiert.





2. Schritt:

Wähle den Masterknoten $\{1.6; 2.6; 3.2; 4.2\} \in \mathbf{M}$.

3. Schritt:

Berechne die Szenarioprobleme (1.6), (2.6), (3.2), (4.2) und erhalte die Lösung

$$y_3^1 = 1,$$
$$x_2^1 = 1,\ x_2^2 = 1,$$
$$y_2^1 = 0,\ y_2^2 = 0,$$
$$y_3^2 = 0,$$
$$x_1^1 = 0,\ x_1^2 = 0,\ x_1^3 = 0,\ x_1^4 = 0,$$
$$y_3^3 = 0,$$
$$x_2^3 = 0,\ x_2^4 = 1,$$
$$y_2^3 = 0,\ y_2^4 = 0,$$
$$y_3^4 = 0,$$

mit dem Zielfunktionswert $\varphi(\{1.6, 2.6, 3.2, 4.2\}) = -0.375$.

4. Schritt:

Alle Szenarioprobleme sind zulässig.

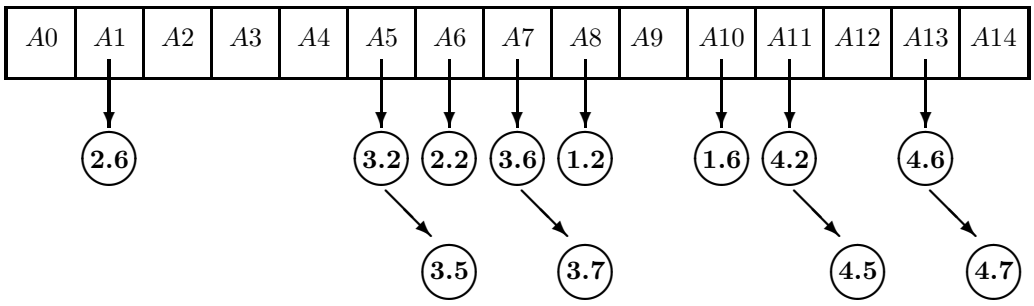
5. Schritt:
Inferiorität ist nicht gegeben.

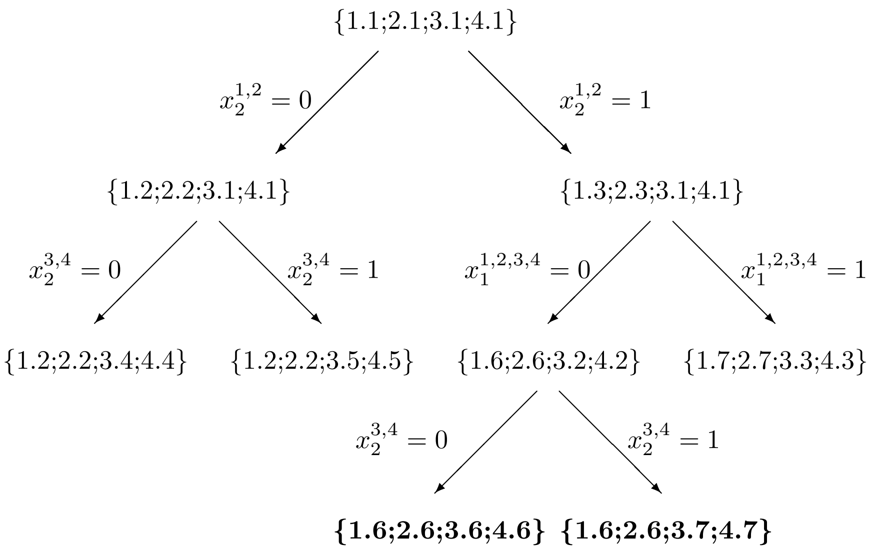
6. Schritt:
Optimalität ist nicht gegeben

7. Schritt:
Erhalte keine zulässige Lösung.

8.Schritt:
Wähle Familie $\mathcal{G}_2^2$ und Variable x_2.
Erzeuge die Szenarioknoten (3.6),(3.7) und (4.6), (4.7), und überprüfe ob diese bereits in **H** enthalten sind. Falls nicht berechne die Adressen und ordne sie entsprechend ein.

Iteration 8:
Wiederum kommt es zu Einspareffekten bei den Szenarioknoten, da in den neu erzeugten Masterknoten die Szenarioknoten (1.6) und (2.6) bereits berechnet wurden.





2. Schritt:

Wähle den Masterknoten $\{1.6; 2.6; 3.6; 4.6\} \in \mathbf{M}$.

3. Schritt:

Berechne die Szenarioprobleme $(3.4), (4.4)$, übernehme $(1.6), (2.6)$ und erhalte die Lösung

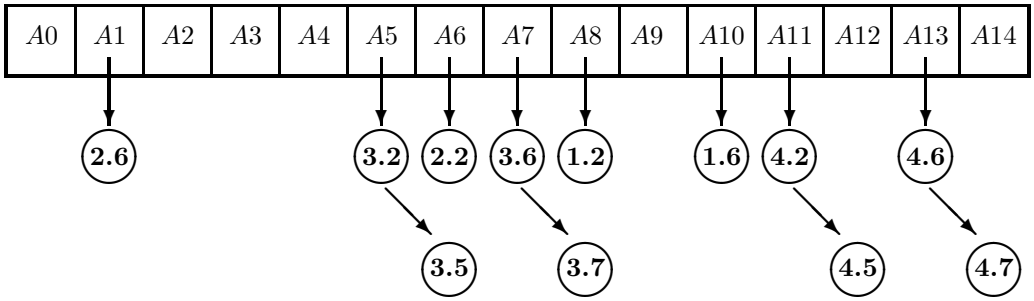
$$y_3^1 = 1,$$
$$x_2^1 = 1,\ x_2^2 = 1,$$
$$y_2^1 = 0,\ y_2^2 = 0,$$
$$y_3^2 = 0,$$
$$x_1^1 = 0,\ x_1^2 = 0,\ x_1^3 = 0,\ x_1^4 = 0,$$
$$y_3^3 = 0,$$
$$x_2^3 = 0,\ x_2^4 = 0,$$
$$y_2^3 = 0,\ y_2^4 = 0,$$
$$y_3^4 = 0,$$

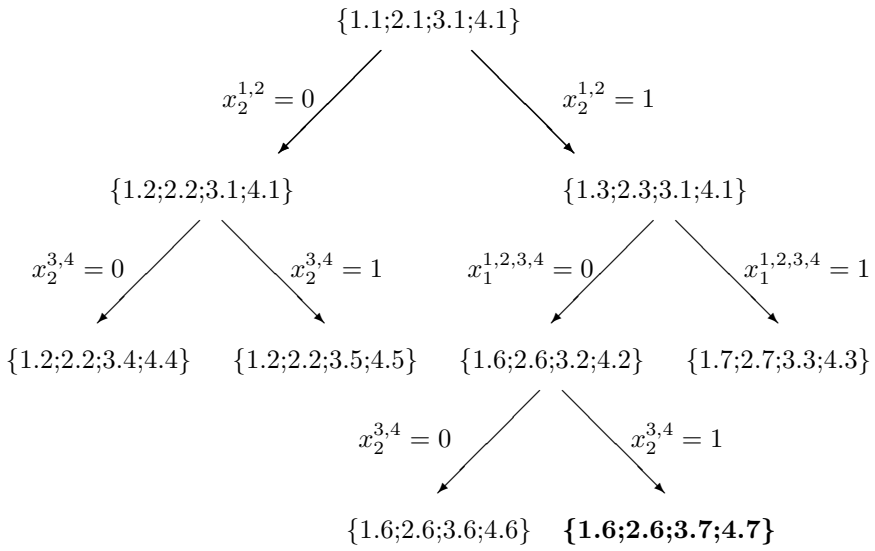
mit dem Zielfunktionswert $\varphi(\{1.6, 2.6, 3.6, 4.6\}) = -0.125$.

4. Schritt:
Alle Szenarioprobleme sind zulässig.

5. Schritt:
Inferiorität ist gegeben → gehe zu Schritt 2.

Iteration 9:
Der Masterknoten wurde gelöscht.





2. Schritt:
Wähle den Masterknoten $\{1.6; 2.6; 3.7; 4.7\} \in \mathbf{M}$.

3. Schritt:
Berechne die Szenarioprobleme $(3.5), (4.5)$, übernehme $(1.6), (2.6)$ und erhalte die Lösung

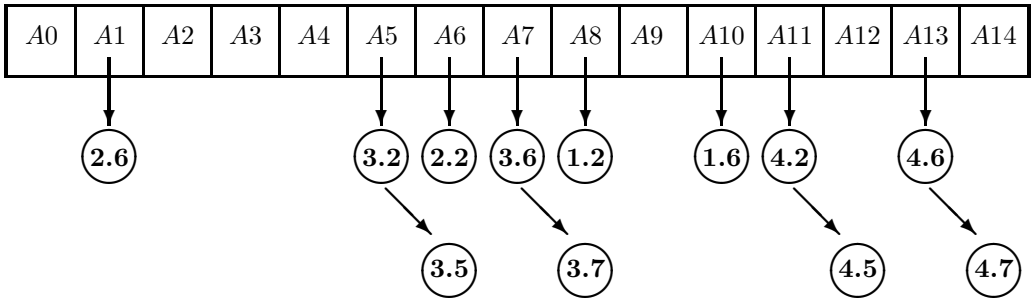
$$\begin{aligned} & & & & & & & & & & & & & & y_3^1 = 1, \\ & & & & & & & & & & & & x_2^1 = 1,\ x_2^2 = 1, & & \\ & & & & & & & & & & & & y_2^1 = 0,\ y_2^2 = 0, & & \\ & & & & & & & & & & & & & & y_3^2 = 0, \\ x_1^1 = 0,\ x_1^2 = 0,\ x_1^3 = 0,\ x_1^4 = 0, & & & & & & & & & & & & & & \\ & & & & & & & & & & & & & & y_3^3 = 1, \\ & & & & & & & & & & & & x_2^3 = 1,\ x_2^4 = 1, & & \\ & & & & & & & & & & & & y_2^3 = 0,\ y_2^4 = 0, & & \\ & & & & & & & & & & & & & & y_3^4 = 0, \end{aligned}$$

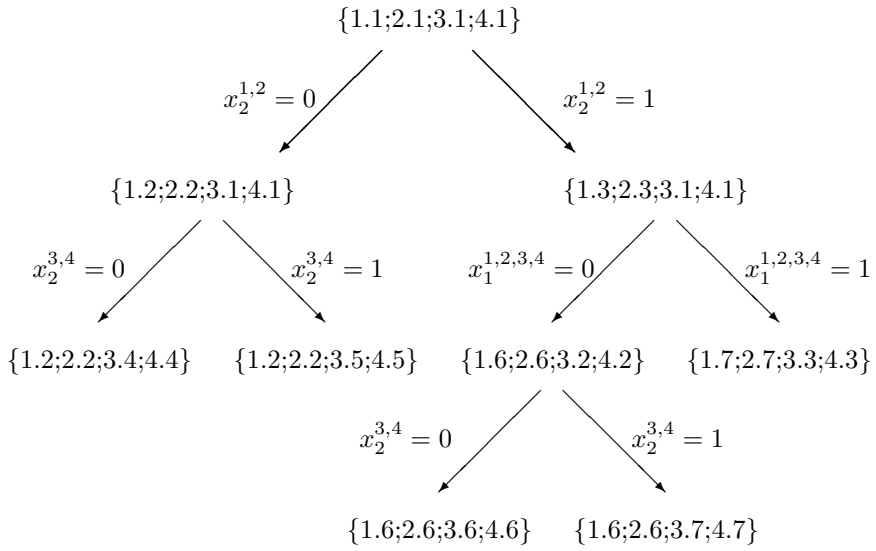
mit dem Zielfunktionswert $\varphi(\{1.6, 2.6, 3.7, 4.7\}) = -0.25$.

4. Schritt:
Alle Szenarioprobleme sind zulässig.

5. Schritt:
Inferiorität ist gegeben → gehe zu Schritt 2.

Iteration 10:





2. Schritt:
$\mathbf{M} = \emptyset \rightarrow$ STOPP. $z = -0.25$ ist optimal.

In diesem Beispiel wurden ebenfalls 33% weniger Szenarioknoten berechnet als in der Basisversion des Algorithmus in Kapitel 3.3. Wir werden in Kapitel 4 sehen, dass auch bb-hash enormes Einsparpotenzial an zu berechnenden Szenarioknoten bei größeren Problemen birgt.

3.3.3 Vergleich

Wir wollen in diesem Abschnitt noch einmal die wesentlichen Unterschiede zwischen bb-tree und bb-hash zusammenfassen und deren Auswirkungen diskutieren. Des Weiteren werden wir diesbezüglich einen Ausblick auf die in Kapitel 4 dokumentierten Rechenergebnisse geben.

Die Algorithmen unterscheiden sich im Wesentlichen in den folgenden drei Punkten:

i) Das Auffinden der Szenarioknoten.

ii) Das Löschen der Szenarioknoten.

iii) Die Wahl der Branchingentscheidung.

Beim Auffinden der Szenarioknoten ist bb-tree durch die binäre Baumstruktur der Knotenverwaltung gegenüber bb-hash eindeutig im Vorteil. Zwar birgt die Wahl der Hash-Funktion einigen Spielraum zur Optimierung, die Geschwindigkeit eines direkten Zugriffs kann jedoch auf keinen Fall erreicht werden. Inwieweit sich der Nachteil der Suche nach Szenarioteilproblemen auf die gesamte Rechenzeit auswirkt, hängt entscheidend von der Zeit ab, die zum Lösen der Teilprobleme benötigt wird. Das liegt in der Unabhängigkeit der Suche von der Größe der Probleme. Je mehr Zeit die Berechnung in Anspruch nimmt, desto geringer ist der relative Anteil an Zeit, der für das Auffinden der Teilprobleme benötigt wird.

Im zweiten Punkt liegen die Vorteile ebenfalls bei bb-tree. Zum Einen werden durch die Systematik nur Szenarioknoten gelöscht, die im weiteren Verlauf nicht mehr benötigt werden, zum Anderen wird kein Eingabeparameter verlangt, der der Hardware angepasst werden muss. Dieser Parameter, der in bb-hash festlegt, in welcher Iteration die zur Zeit nicht benötigten Szenarioknoten gelöscht werden sollen, darf erfahrungsgemäß nicht zu groß gewählt werden, da ansonsten Speicherkapazitäten überschritten werden. Kommt es auf der anderen Seite zu einem häufigen Abschneiden der Szenarioknoten, treten oft Mehrfach-Berechnungen auf.

Nach den Erfahrungen, die durch die Testrechnungen in Kapitel 4 gewonnen wurden, führt der dritte Punkt zu den deutlichsten Unterschieden zwischen beiden Algorithmen. Durch die fest vorgegebene Branchingstrategie in bb-tree können die Masterknoten zwar nicht individuell behandelt werden, das Verfahren hat jedoch den Vorteil, dass Szenarioteilprobleme verhältnismäßig häufig benutzt werden. Da die Branchingentscheidung in bb-hash frei wählbar ist, entstehen des öfteren Teilprobleme, die in nur sehr wenigen Masterknoten enthalten sind. Dieser Sachverhalt schlägt sich in einem proportional höheren Anteil an berechneten Szenarioknoten, im Vergleich zur Anzahl an Masterknoten nieder, siehe hierzu Kapitel 4.2.2. Anders ausgedrückt werden mit der gleichen Anzahl an Szenarioknoten in bb-tree mehr Ma-

sterknoten erzeugt. In bb-hash erhält man bei einer gleichen Anzahl an Masterknoten im Allgemeinen bessere untere Schranken, was auf die individuelle Behandlung zurückzuführen ist.

Kapitel 4

Anwendungen

In diesem Kapitel werden wir zwei Anwendungen vorstellen. Es handelt sich zum einem um ein praxisorientiertes Problem aus der elektrischen Energieversorgung und zum anderen um das eher akademische Knapsack Problem.

Bei beiden Anwendungen werden wir die Rechengeschwindigkeit der Algorithmen bb-tree und bb-hash mithilfe von Testrechnungen dokumentieren. Zum Vergleich wird das wohl weit verbreitetste kommerziell verfügbare Programm zum Lösen gemischt-ganzzahliger Optimierungsprobleme, CPLEX [57] genutzt.

Zusätzlich wird bei der ersten Anwendung auf konzeptionelle Aspekte durch die Nutzung verschiedener Mean-Risk Modelle eingegangen. Es wird anhand überschaubarer Problemgrößen gezeigt, welchen Einfluss die Hinzunahme der Risikomaße auf die Lösung hat.

Ein interner Vergleich von bb-tree und bb-hash, der insbesondere das Einsparpotential an zu berechnenden Szenarioproblemen dokumentiert, wird mithilfe des Knapsack Problems vorgenommen.

In allen Rechnungen wurde eine C-Implementierung von bb-tree und bb-hash genutzt. Diese verwendet das von H.I. Gassmann und E. Schweitzer entwickelte SMPS-Format [39, 40] zur Dateneingabe. Die Teilprobleme werden bei beiden Algorithmen mit CPLEX [57] gelöst. Eine Bedienungsanleitung findet man in [48] bzw. [49].

4.1 Elektrische Versorgungsnetze

Die derzeitige elektrische Energieversorgung beruht auf einer zentralen Struktur, in der konventionelle Großkraftwerke in Hochspannungsnetze einspeisen. Die steigende Nachfrage nach erneuerbaren Energien führte zu einem verstärkten Interesse an mathematischer Modellierung und Analyse dezentraler Energieversorgungssysteme, siehe etwa [42, 46, 66]. Charakteristisch für dieses Versorgungssystem ist nach [2], dass der Standort und die Einspeisung der Energieumwandlungsanlagen in unmittelbarer Nähe des Bedarfs liegt und ihre Nennleistung kleiner als fünf MW ist.

Durch die zufallsabhängige Verfügbarkeit natürlicher Ressourcen wie Wind- oder Sonnenenergie, sowie einer zufallsabhängigen Nachfrage, ist eine realistische Modellbildung nur unter Berücksichtigung stochastischer Eingabedaten möglich. Zudem führen An/Aus-Entscheidungen zu unvermeidbaren Ganzzahligkeitsanforderungen, so dass die stochastische, lineare, gemischt-ganzzahlige Optimierung ein geeignetes Werkzeug zum Lösen solcher Probleme darstellt.

Arbeiten zur elektrischen Energieversorgung bei unsicherer Nachfrage findet man unter anderem in [17, 83]. In [116] wird ein Überblick über stochastische Optimierungsmethoden in diesem Anwendungsgebiet gegeben.

Wir werden das in [66] entwickelte Modell betrachten. Die Autoren benutzen zum Lösen dieses Problems einen zweistufigen Ansatz. In dieser Arbeit wird das mehrstufige Kompensationsmodell angewendet.

4.1.1 Problembeschreibung

Wir betrachten elektrische Versorgungssysteme, wie in Abbildung 4.1. In diesem Beispiel hat das System sechs Entnahmestationen. Ins Netz eingespeist wird in den ersten vier Stationen. Da das Speichern elektrischer Energie sehr kostspielig und ineffizient ist, besteht die Aufgabe darin, eine optimale Strategie zu berechnen, die die Nachfrage deckt und eine Überproduktion möglichst vermeidet. In vielen Modellen wird das Netzwerk vernachlässigt, indem das gesamte Angebot und die gesamte Nachfrage in jeweils einer Station zusammengefasst wird. Wir werden in unserem Modell auf diese Vereinfachung verzichten und die zusätzlichen technischen Restriktionen und Kapazitätsbeschränkungen mit berücksichtigen.

Das elektrische Versorgungsnetz wird hier durch einen ungerichteten Graphen $G = (V, E)$ modelliert. Die Knoten V repräsentieren die Einspeise- bzw. Entnahmestationen. Die Kantenmenge E stellt die Verbindungen zwischen den Stationen dar. Jede Verbindung ist charakterisiert durch ihre Kapazität und Leitfähigkeit. Eine umfassende Einführung in die Modellierung elektrischer Versorgungsnetze ist in [7, 45] gegeben.

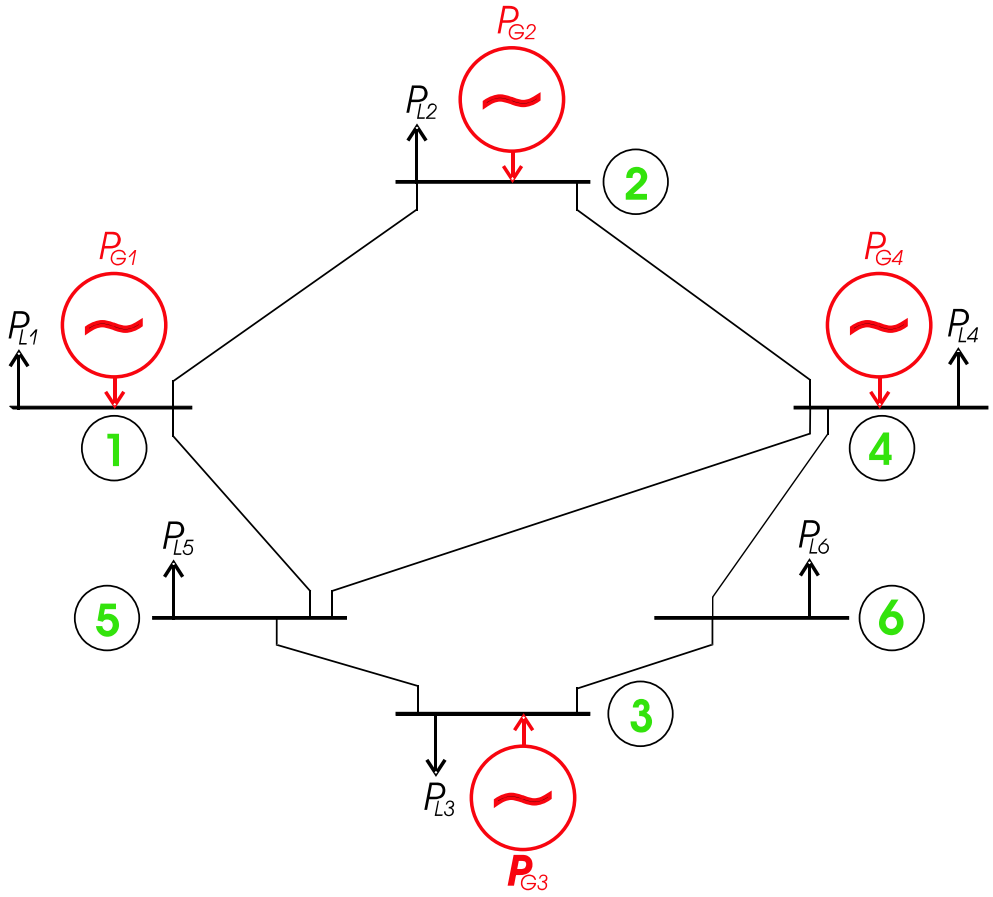


Abbildung 4.1: Elektrisches Versorgungssystem

In Netzwerken mit Wechselstrom wird die elektrische Leistung mithilfe komplexer Zahlen modelliert, wobei der Realteil die Wirkleistung und der Imaginärteil die Blindleistung repräsentieren. Dadurch ist für jeden Knoten $k \in V$ die komplexe Leistung gegeben durch $S^k = P^k + i\,Q^k$, welche zu den Leistungsflussgleichungen

$$P^k = \sum_{l \in \mathcal{N}^k} p^{kl}$$

$$Q^k = \sum_{l \in \mathcal{N}^k} q^{kl}$$

führen. Hier ist $\mathcal{N}^k \subseteq V$ die Menge der zu $k \in V$ adjazenten Knoten. Die Variablen p^{kl}, q^{kl} repräsentieren den Real- und Imaginärteil des Leistungsflusses entlang der Kante $kl \in E$ und müssen folgende Gleichungen erfüllen:

$$p^{kl} = (U^k)^2 g^{kl} - U^k U^l g^{kl} \cos\theta^{kl} - U^k U^l b^{kl} \sin\theta^{kl} \tag{4.1}$$

$$q^{kl} = U^k U^l b^{kl} \cos\theta^{kl} - U^k U^l g^{kl} \sin\theta^{kl} - (U^k)^2 (b^{kl} + b_0^{kl}). \tag{4.2}$$

U^k ist der absolute Teil der komplexen Spannung $U^k\, e^{i\theta^k}$ im Knoten $k \in V$, θ^k der Phasenwinkel und θ^{kl} die Phasenwinkeldifferenz der Knoten $k, l \in V$. Die Größen g^{kl}, b^{kl}, b_0^{kl} sind Konstanten, die die Leitfähigkeit der Kante $kl \in E$ beschreiben. Diese Gleichungen führen zu einem nicht-konvexen Optimierungsproblem, welches mit dem heutigen Kenntnisstand kaum zu lösen ist. Wir werden deshalb die folgenden Vereinfachungen vornehmen: In Hochspannungsnetzwerken ist die Annahme $\theta^{kl} \approx 0$ zulässig. Dieses führt zu $\cos\theta^{kl} \approx 1$ und $\sin\theta^{kl} \approx \theta^{kl}$. Des Weiteren kann $U^k = 1$ gesetzt werden, siehe [45]. Somit geht (4.1) über in

$$p^{kl} = b^{kl}(\theta^l - \theta^k). \tag{4.3}$$

Eine weitere Vereinfachung ist die Vernachlässigung der Blindleistung. Damit reduziert sich das Gleichungssystem (4.1), (4.2) auf (4.3), welches man als Leistungsflussgleichungssystem in Gleichstrom-Formulierung bezeichnet.

Mit diesen Vereinfachungen entwickeln wir nun das folgende lineare gemischtganzzahlige Modell. Wir betrachten einen diskreten Planungshorizont von T Zeitschritten $\{1, \ldots, T\}$, welcher in gleichgroße Zeitabschnitte unterteilt ist. Die Menge $V' \subseteq V$ sei die Menge aller Knoten, in denen Energieeinspeisung möglich ist. Die Variable $P_t^{G,k} \in \mathbb{R}_+$ sei die eingespeiste Energie in Knoten k zum Zeitpunkt t und die Daten $P_t^{L,k} \in R_+$ die Nachfrage in Knoten k zum Zeitpunkt t. Mit dieser Notation erhalten wir die Leistungsflussgleichungen:

$$P_t^{G,k} - P_t^{L,k} = \sum_{l \in \mathcal{N}_k} p_t^{kl} \qquad \forall t \in \{1, \ldots, T\}\ \forall k \in V' \tag{4.4}$$

$$-P_t^{L,k} = \sum_{l \in \mathcal{N}^k} p_t^{kl} \qquad \forall t \in \{1, \ldots, T\}\ \forall k \in V' \setminus V \tag{4.5}$$

$$p_t^{kl} = b^{kl}(\theta_t^l - \theta_t^k) \qquad \forall t \in \{1, \ldots, T\}\ \forall kl \in E. \tag{4.6}$$

Da in (4.6) nur die Differenz der Phasenwinkel von Interesse ist, macht es Sinn, diese in einem sogenannten Referenzknoten, z.B. für $k = 1$ gleich Null zu setzen:

$$\theta_t^1 = 0 \quad \forall t \in \{1, \ldots, T\}. \tag{4.7}$$

Um eine Überbeanspruchung des Netzwerks und damit einen Zusammenbruch der Versorgung zu verhindern, müssen die Gleichungen

$$-p_{\max}^{kl} \leq p_t^{kl} \leq p_{\max}^{kl} \quad \forall t \in \{1, \ldots, T\} \; \forall kl \in E \tag{4.8}$$

eingehalten werden, wobei $p_{\max}^{kl}$ die maximale Kapazität der Leitung $kl \in E$ ist.

Der Schaltzustand der Einspeisestationen bzw. des Kraftwerkes in einem Knoten $k \in V'$, zu einem Zeitpunkt t, wird mit den Variablen $s_t^k \in \{0, 1\}$ beschrieben. Hier induziert $s_t^k = 1$, dass das entsprechende Kraftwerk zum Zeitpunkt t in Betrieb ist. Damit muss bei der Einspeisung folgende Gleichung berücksichtigt werden:

$$P_{\min}^{G,k} \cdot s_t^k \leq P_t^{G,k} \leq P_{\max}^{G,k} \cdot s_t^k \quad \forall t \in \{1, \ldots, T\} \; \forall k \in V', \tag{4.9}$$

wobei $P_{\min}^{G,k}$ und $P_{\max}^{G,k}$ der minimale- bzw. maximale Ausstoß im Knoten k ist.

Mit den Leistungsgradienten Δ_{hoch}^k und Δ_{runter}^k wird beschrieben, wie stark man den Leistungsaustoss des Kraftwerks in Knoten k zwischen zwei Zeitintervalle maximal erhöhen bzw. verringern kann. Die Restriktionen

$$P_t^{G,k} - P_{t-1}^{G,k} \leq m\,\Delta_{hoch}^k \quad \forall t \in \{1, \ldots, T\} \; \forall k \in V' \tag{4.10}$$

$$P_{t-1}^{G,k} - P_t^{G,k} \leq m\,\Delta_{runter}^k \quad \forall t \in \{1, \ldots, T\} \; \forall k \in V' \tag{4.11}$$

spiegeln diesen Sachverhalt wieder.

Für alle Zeitstufen $t \in \{1, \ldots, T\}$ und Knoten $k \in V'$ definieren wir Schaltvariablen $u_t^k, v_t^k \in \{0, 1\}$, wobei $u_t^k = 1$, falls das k-te Kraftwerk zum Zeitpunkt t angeschaltet und $v_t^k = 1$, falls das k-te Kraftwerk zum Zeitpunkt t ausgeschaltet wird. Es müssen also folgende Bedingungen eingehalten werden:

$$s_t^k - s_{t-1}^k = u_t^k - v_t^k \quad \forall t \in \{1, \ldots, T\} \; \forall k \in V' \tag{4.12}$$

$$u_t^k + v_t^k \leq 1 \quad \forall t \in \{1, \ldots, T\} \; \forall k \in V'. \tag{4.13}$$

Mithilfe dieser Schaltvariablen werden Mindestlauf- bzw. Mindeststillstandszeiten $(\Lambda_{lz}^k, \Lambda_{sz}^k)$ in dem Modell berücksichtigt.

$$s_{t+n}^k \geq u_t^k \quad \forall t \in \{1, \ldots, T + 1 - \Lambda_{lz}^k\} \; \forall n \in \{1, \ldots, \Lambda_{lz}^k - 1\} \tag{4.14}$$

$$s_{t+n}^k \leq 1 - v_t^k \quad \forall t \in \{1, \ldots, T + 1 - \Lambda_{sz}^k\} \; \forall n \in \{1, \ldots, \Lambda_{sz}^k - 1\} \tag{4.15}$$

In der Zielfunktion werden Fixkosten c^k, variable Kosten f^k, sowie Anfahrtkosten d^k minimiert.

$$\sum_{t\in\{1,\dots,T\}} \sum_{k\in V'} c^k\ s_t^k + f^k\ P_t^{G,k} + d^k\ u_t^k \longrightarrow \min! \tag{4.16}$$

In obiger Formulierung sind alle Daten deterministisch, was der Natur dieses Problems nur unzureichend Rechnung trägt. Wie schon in der Einführung erwähnt, ist die Nachfrage im Allgemeinen nicht bekannt, so dass wir die Daten $P_t^{L,k}$ als stochastisch voraussetzen. In unserem Grundmodell (3.5) wird dieser Sachverhalt mit zufälligen rechten Seiten berücksichtigt. Die zufallsabhängige Einspeisung von Wind- oder Solarkraftwerken $P_{\max}^{G,k}$ wird in (3.5) durch zufällige Matrixeinträge modelliert. Des Weiteren werden auch die Kapazitätsbeschränkungen des Netzwerkes $p_{\max}^{kl}$ aufgrund von äußeren Witterungsverhältnissen als zufallsabhängig vorausgesetzt. Diese führen ebenfalls zu zufälligen rechten Seiten.

4.1.2 Konzeptionelle Aspekte

In diesem Abschnitt werden wir auf das Lösungsverhalten zwischen dem rein erwartungswertbasierten- und verschiedenen Mean-Risk Modellen eingehen. Wir betrachten hierzu ein IEEE-Testnetz mit insgesamt 14 Stationen, davon fünf Einspeisestationen. Vergleiche hierzu [56]. Der Zeithorizont beträgt sechs Stunden, mit stündlichen Entscheidungspunkten, was zu sechs Zeitstufen führt. Um die Szenariolösungen zu analysieren, betrachten wir eine relativ geringe Anzahl von 16 Szenarien. Wir verwenden für das Mean-Risk Modell die Excess Probability und den Expected Excess mit einem Gewicht von $\beta = 1000$. In den folgenden beiden Rechnungen wurden zwei verschieden Datensätze benutzt.

In den Tabellen 4.1 und 4.2 wurde in der linken Spalte das Erwartungswertmodell berechnet, in der rechten Spalte das Mean-Risk Modell mit dem entsprechenden Risikomaß. Die ersten fünf Zeilen zeigen die Schaltzustände der Einspeisestationen und die sechste Zeile die Kosten der ersten Stufe. In der letzten Zeile stehen der Zielfunktionswert des Erwartungswertmodells, sowie der Erwartungswertanteil des Zielfunktionswertes des Mean-Risk Modells. Die Schaltzustände in der ersten Stufe sind insbesondere dann von großem Interesse, wenn eine sequentielle Optimalitäts-

	$\mathbb{E}$	$\mathbb{E} + \beta \mathcal{R}_{MEP_\varphi}$
s_1^1	1	0
s_1^2	1	1
s_1^3	1	0
s_1^4	0	1
s_1^5	1	0
z_1	2429	3604
EW	13928	13972

Tabelle 4.1: Lösungen und Erwartungswert für verschiedene Modelle

	$\mathbb{E}$	$\mathbb{E} + \beta \mathcal{R}_{MEE_\varphi}$
s_1^1	1	0
s_1^2	0	0
s_1^3	1	1
s_1^4	0	1
s_1^5	1	0
z_1	1672	2784
EW	12534	12574

Tabelle 4.2: Lösungen und Erwartungswert für verschiedene Modelle

berechnung stattfindet: Nach Ausführung der Erststufenentscheidung werden die Daten mit den in Zukunft neu erhaltenen Informationen "aufgefrischt", und das Optimierungsproblem erneut berechnet.

Betrachtet man die Kosten der ersten Stufe, wird der Nutzen der mehrstufigen Modellierung erkennbar. Hätte man die erste Stufe für sich betrachtetet, würde bei dieser Kostendifferenz kein Entscheidungsträger die Alternative, die das Risikomodell vorschlägt in Betracht ziehen. Wie wir anhand der Erwartungswerte sehen, wird diese Kostendifferenz jedoch im Mittel über den gesamten Planungshorizont stark

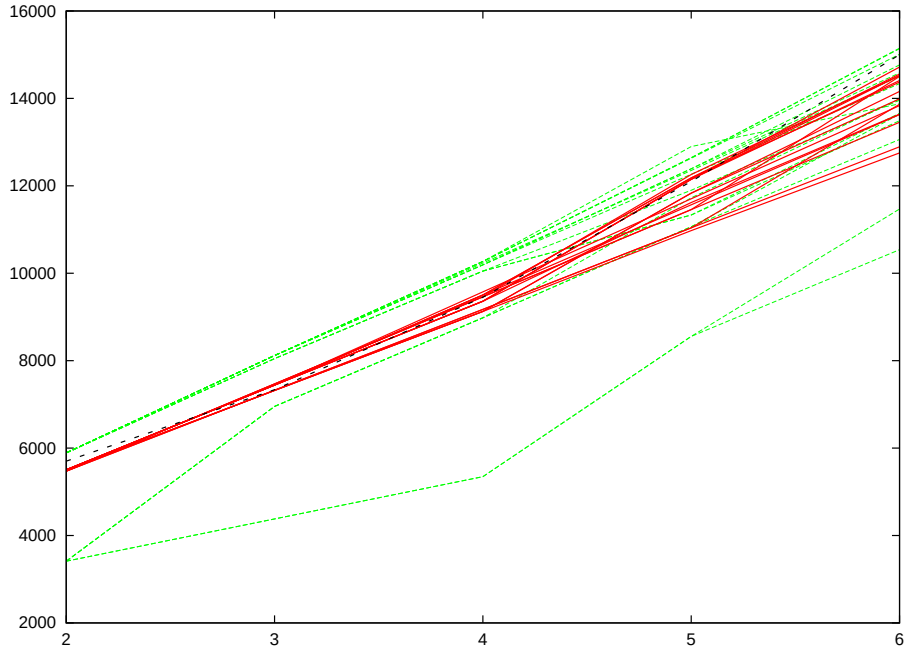


Abb. 4.3: Szenariolösungen bei der Excess Probability

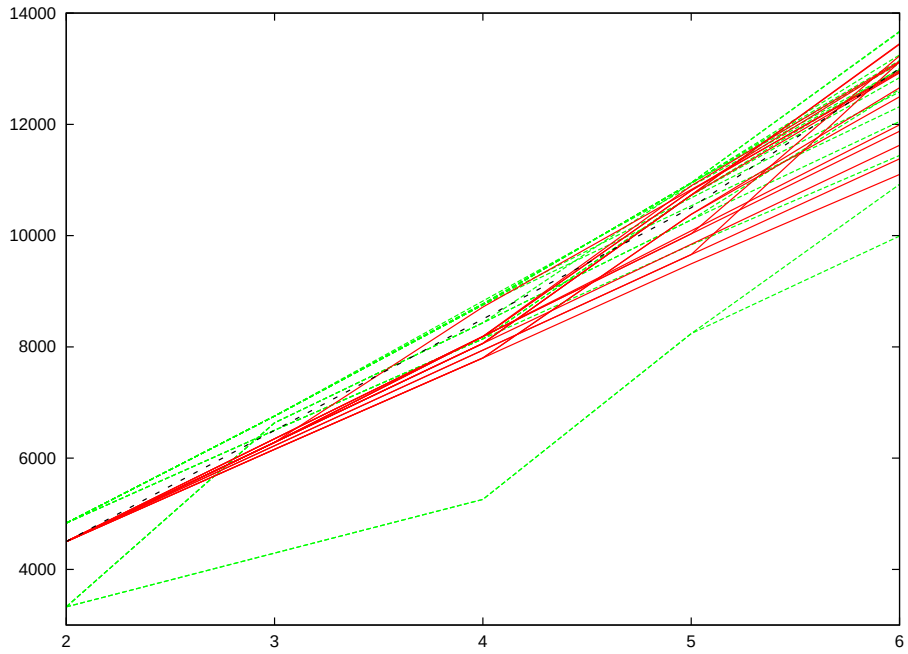


Abb. 4.4: Szenariolösungen beim Expected Excess

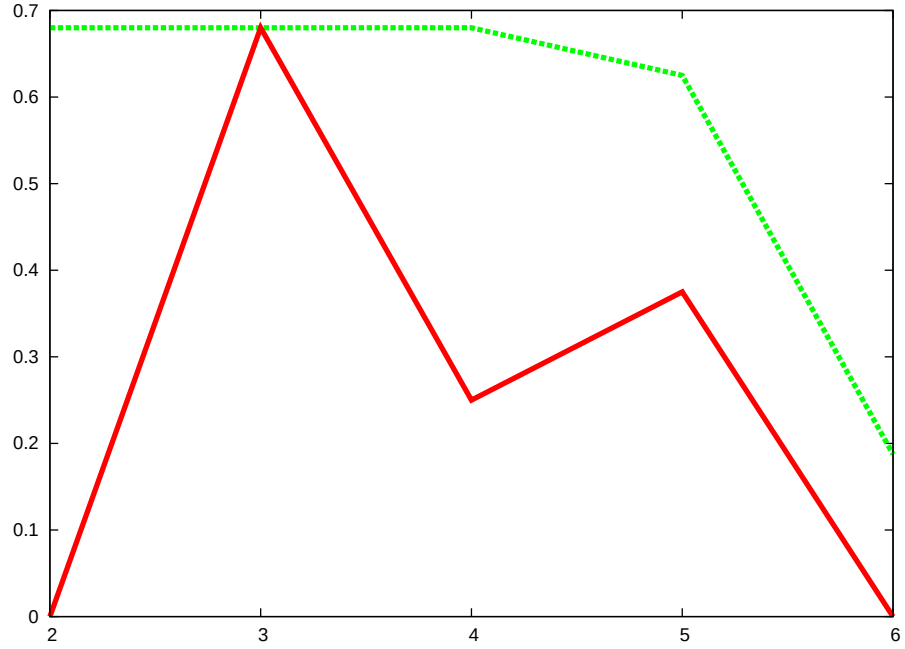


Abb. 4.5: Risikowerte bei der Excess Probability

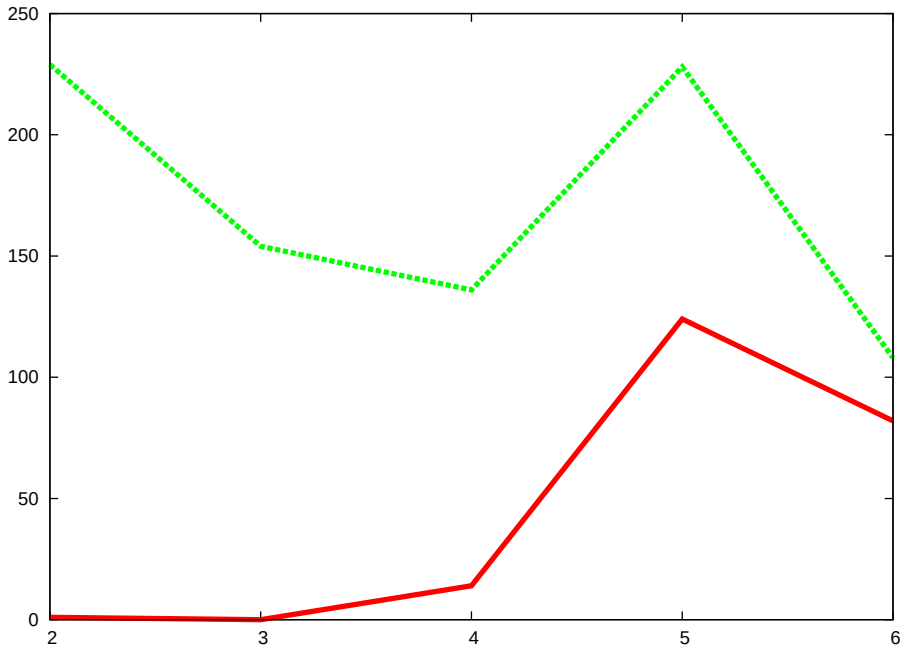


Abb. 4.6: Risikowerte beim Expected Excess

reduziert. Dennoch weist die erste Alternative einen geringeren Erwartungswert auf. Wie sieht es jedoch mit der Streuung der erhaltenen Lösungen aus? In den Abbildungen 4.3 und 4.4 sind die Szenariolösungen der beiden Probleme zu sehen. Die gestrichelten grünen Linien sind die Lösungen der Erwartungswertprobleme. Die durchgezogenen roten Linien die Lösungen der Mean-Risk Probleme. Die schwarzen gestrichelten Linien deuten die verwendeten Schranken φ_t, $t = 2, \ldots, 6$ an. Man sieht deutlich, dass unter Verwendung der Risikomaße die Streuung signifikant reduziert werden kann. Die konkreten Werte der Risikomaße für alle Zeitstufen sind in den Abbildungen 4.5 und 4.6 zu sehen. Wiederum veranschaulichen die grünen gestrichelten Linien die Ergebnisse der Erwartungswertprobleme, die roten durchgezogenen Linien die Ergebnisse der Mean-Risk Probleme. Insgesamt steht einer relativ kleinen Verschlechterung des Erwartungswertes eine deutliche Reduzierung des Risikowertes in fast allen Zeitstufen gegenüber.

Wie bereits erwähnt, benutzen wir bei der Optimierung die akkumulierten Kosten, vergleiche auch Kapitel 2.2. Ein weiterer interessanter Aspekt ist die Auswirkung der Modelle auf die Kosten in den einzelnen Zeitschritten. Siehe hierzu die

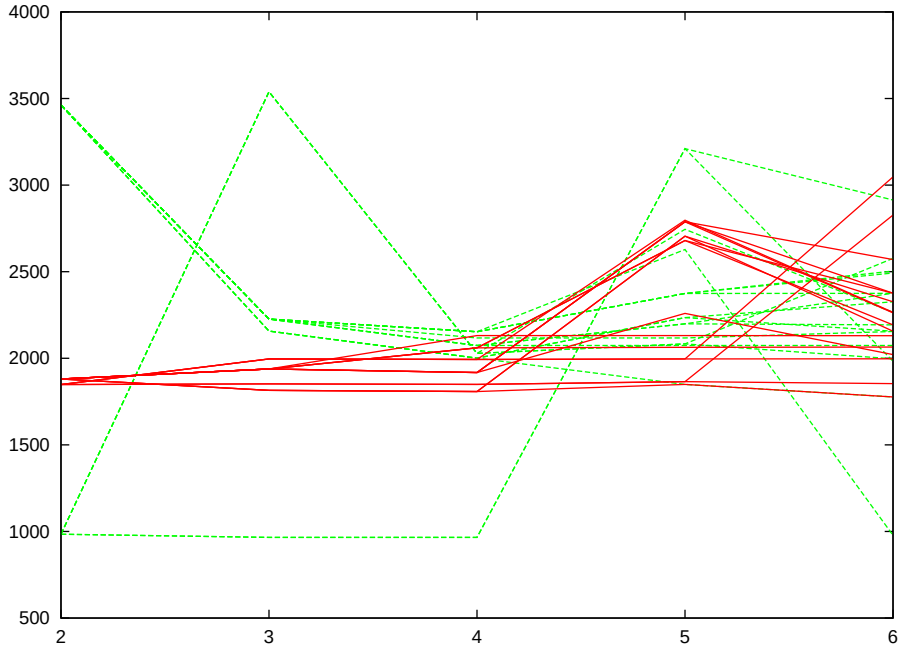


Abb. 4.7: Szenariolösungen bei der Excess Probability

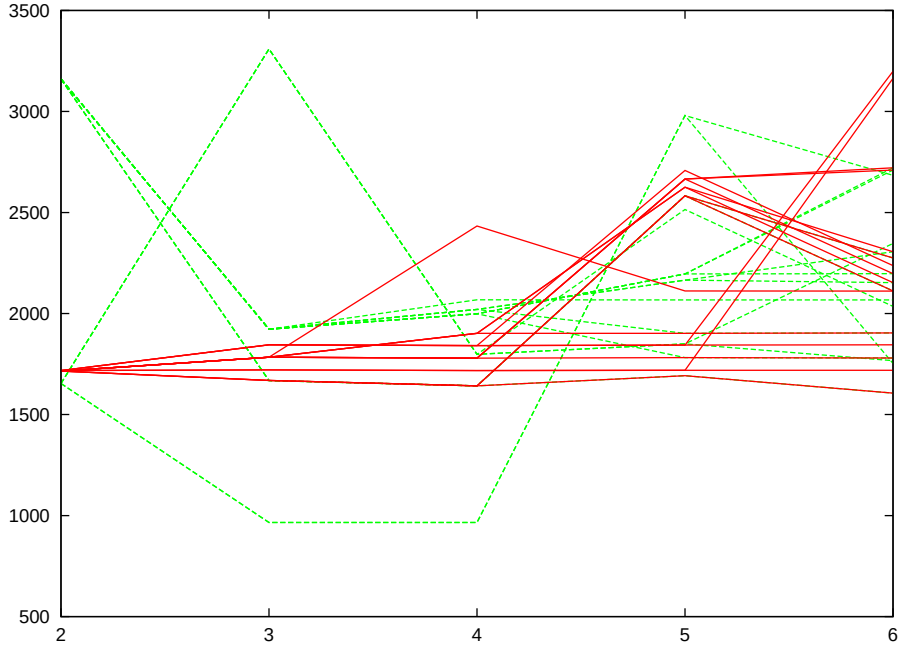


Abb. 4.8: Szenariolösungen beim Expected Excess

Abbildungen 4.7 und 4.8. Es wurden hier die akkumulierten Kosten optimiert und anhand der berechneten Lösungen die Kosten in den einzelnen Zeitstufen berechnet. Obwohl diese Kosten nicht explizit im Kontext der Risikominimierung optimiert wurden, erhalten wir ebenfalls eine starke Reduzierung der Streuung.

4.1.3 Testrechnungen

In diesem Abschnitt wird die Rechengeschwindigkeit von bb-tree und bb-hash mit CPLEX Version 9.1.3 verglichen. Alle Probleme wurden auf einem Sun Ultra Sparc III mit einem 1.2 GHz Prozessor und 32 GB RAM Arbeitsspeicher berechnet.

Wie im vorherigen Abschnitt verwenden wir für unsere Testprobleme das 14-Bus IEEE-Testnetz [56]. Die Tabellen 4.9, 4.10 und 4.11 veranschaulichen die Resultate. In der ersten Spalte stehen die Bezeichnungen der verschiedenen Testinstanzen. Die zweite Spalte gibt die Anzahl der Zeitstufen an; die dritte Spalte die Anzahl der Szenarien. In der vierten und fünften Spalte wird dokumentiert, wie groß die entstandenen Probleme sind. Wir unterscheiden hierbei zwischen der Anzahl der

stetigen und der binären Variablen. Wir berechneten Mean-Risk Modelle mit den Risikomaßen Excess Probability, Expected Excess und Conditional Value-at-Risk. Die Gewichte sind der sechsten Spalte zu entnehmen, wobei im Fall $\beta = 0$ nur der Erwartungswert optimiert wurde. Die Gewichte deuten eine Gleich- bzw. Übergewichtung des Risikomaßes an. Eine Angleichung bei verschiedenen Risikomaßen - bei der Excess Probability sind die Werte wesentlich kleiner als beim Expected Excess - wurde durch die Gewichte γ_t, $t = 2, \ldots, T$ vorgenommen.

Bei den Berechnungen wurde eine Zeitschranke von 30 Minuten und ein relatives Gap (RG) von 0.1% als Abbruchkriterium gewählt. Das relative Gap wurde definiert durch

$$RG := \frac{UB - LB}{|UB|} \cdot 100\%, \tag{4.17}$$

wobei UB die aktuell kleinste obere- und LB die aktuell kleinste untere Schranke ist. In den Spalten sieben und acht wurde bb-tree mit CPLEX verglichen. Diese beiden Spalten sind folgendermaßen zu verstehen: Bei dem Programm, das zuerst ein relatives Gap von 0.1% erreichte, wurde die benötigte Zeit in Sekunden notiert, bei dem anderen Programm wurde das relative Gap eingetragen, das bis zu diesem Zeitpunkt erreicht wurde. Konnten beide Programme nicht innerhalb einer halben Stunde das vorgegebene Gap erreichen, wurde das bis dahin erhaltene Gap notiert. In den Spalten neun und zehn wurde auf die gleiche Weise bb-hash mit CPLEX verglichen.

Wir sehen, dass die Ergebnisse von bb-tree und bb-hash bei der Excess Probability und dem Expected Excess ausnahmslos besser sind. Mit bb-tree konnte für alle Instanzen das vorgegebene Gap innerhalb von 30 Minuten erreicht werden. Auch bei den etwas durchwachsenen Ergebnissen beim Conditional Value-at-Risk kann festgehalten werden, dass bb-tree und bb-hash bei mehr als der Hälfte der Beispielprobleme bessere Ergebnisse lieferte als CPLEX.

Test	T	S	Variablen Stet/Bin	Nebenbe-dingungen	β	bb-tree	CPLEX	bb-hash	CPLEX
EP1			4189/1065	7450	0	66s	3.5%	135s	3.2%
EP2	3	50	4189/1135	7520	1	71s	4.7%	137s	3.8%
EP3			4189/1135	7520	2	72s	6.1%	130s	5.3%
EP4			7139/1815	13200	0	358s	9.5%	1376s	8.1%
EP5	3	100	7139/1935	13320	1	405s	11.6%	4.7%	9.2%
EP6			7139/1935	13320	2	387s	9.3%	1.2%	7.8%
EP7			6254/1590	13076	0	160s	6.1%	315s	5.3%
EP8	5	50	6254/1695	13181	1	129s	6.2%	309s	4.7%
EP9			6254/1695	13181	2	114s	6.7%	284s	4.5%
EP10			10974/2790	24026	0	404s	6.0%	684s	5.3%
EP11	5	100	10974/2975	24211	1	290s	7.0%	327s	6.9%
EP12			10974/2975	24211	2	210s	20.1%	317s	12.4%

Tabelle 4.9: Excess Probability - Vergleich mit CPLEX

Test	T	S	Variablen Stet/Bin	Nebenbe-dingungen	β	bb-tree	CPLEX	bb-hash	CPLEX
EE1			4189/1065	7450	0	66s	3.5%	135s	3.2%
EE2	3	50	4259/1065	7520	1	63s	4.2%	127s	3.5%
EE3			4259/1065	7520	2	61s	3.4%	129s	2.8%
EE4			7139/1815	13200	0	358s	9.5%	1376s	8.1%
EE5	3	100	7259/1815	13320	1	406s	10.9%	1.7%	7.9%
EE6			7259/1815	13320	2	405s	8.7%	1.0%	8.4%
EE7			6254/1590	13076	0	160s	6.1%	315s	5.3%
EE8	5	50	6359/1590	13181	1	141s	4.9%	269s	4.4%
EE9			6359/1590	13181	2	97s	9.5%	228s	5.8%
EE10			10974/2790	24026	0	404s	6.0%	684s	5.3%
EE11	5	100	11159/2790	24211	1	339s	5.9%	273s	6.2%
EE12			11159/2790	24211	2	239s	13.1%	267s	13.0%

Tabelle 4.10: Expected Excess - Vergleich mit CPLEX

Test	T	S	Variablen Stet/Bin	Nebenbe-dingungen	β	bb-tree	CPLEX	bb-hash	CPLEX
CV1			4189/1065	7450	0	66s	3.5%	135s	3.2%
CV2	3	50	4261/1065	7520	1	1647s	1.7%	0.4%	1.6%
CV3			4261/1065	7520	2	4.4%	1.7%	4.4%	1.7%
CV4			7139/1815	13200	0	358s	9.5%	1376s	8.1%
CV5	3	100	7261/1815	13320	1	4.1%	8.0%	4.4%	8.0%
CV6			7261/1815	13320	2	3.5%	7.9%	3.7%	7.9%
CV7			6254/1590	13076	0	160s	6.1%	315s	5.3%
CV8	5	50	6363/1590	13181	1	4.7%	2.7%	4.7%	2.7%
CV9			6363/1590	13181	2	1.8%	3.2%	1.7%	3.2%
CV10			10974/2790	24026	0	404s	6.0%	684s	5.3%
CV11	5	100	11163/2790	24211	1	2.5%	4.5%	10.2%	4.5%
CV12			11163/2790	24211	2	3.3%	4.5%	5.9%	4.5%

Tabelle 4.11: Conditional Value-at-Risk - Vergleich mit CPLEX

4.2 Multi-Knapsack-Probleme

Knapsack-Probleme gehören zu den klassischen Problemen der kombinatorischen Optimierung. Erste Arbeiten gehen auf das 19. Jahrhundert zurück, siehe etwa [77]. In seiner ursprünglichsten Form besteht das Problem darin, aus einer Anzahl von n Gegenständen mit positiven Gewichten und Vergütungen, eine optimale Auswahl derart zu treffen, dass vorgegebene Kapazitäten nicht überschritten werden:

$$\max \left\{ \sum_{j=1}^{n} c_j \, x_j : \sum_{j=1}^{n} a_j \, x_j \leq b, \; x_j \in \{0,1\}, \; j = 1, \ldots, n \right\}. \tag{4.18}$$

Einen rigorosen Überblick über verschiedene Modifikationen und populäre Algorithmen bietet das Buch [63].

4.2.1 Problembeschreibung

Bei unseren Rechnungen betrachten wir ein Multi-Knapsack-Problem. Im Gegensatz zu (4.18) sind dabei m Kapazitätsrestriktionen einzuhalten. Zusätzlich sind

die Entscheidungsvariablen nicht auf den binären Fall beschränkt, so dass mehrere Einheiten des selben Gegenstandes gewählt werden können.

Der dynamische Ansatz wird dadurch induziert, dass in dem gegebenen diskreten Zeithorizont $1, \ldots, T$ zu jedem Zeitpunkt $t \in \{1, \ldots, T-1\}$ ein Problem mit den Knapsack-Restriktionen

$$A_t^1 \, x_t \leq b_{t+1}, \; x_t \in \mathbb{Z}_+^{n_t}, \tag{4.19}$$

gelöst wird. Die Anzahl der Gegenstände n_t und der Restriktionen m_t können in Abhängigkeit vom Zeitpunkt $t \in \{1, \ldots, T-1\}$ gewählt werden. Diese zeitgebundenen Entscheidungen werden durch Nebenbedingungen der Form

$$\sum_{\tau=0}^{t-1} A_{t,\tau}^2 \, x_{t-\tau} \leq h_t, \;\; t = 1, \ldots, T-1. \tag{4.20}$$

miteinander gekoppelt. Die stochastische Erweiterung erfolgt durch zufallsabhängige rechte Seiten $b_t, t = 1, \ldots, T-1$ der Knapsack-Restriktionen. Eventuelle Überschüsse zum Zeitpunkt $t \in \{1, \ldots, T-1\}$ werden durch Strafkosten $d_{t+1} \leq 0$ in der folgenden Zeitperiode kompensiert. Modelliert wird dieses, indem (4.19) mithilfe zusätzlicher Variablen $y_{t+1} \in \{0,1\}^{m_t}$, $t = 1, \ldots, T-1$ zu

$$A_t^1 x_t \leq b_{t+1} + \mathbf{1} \, M \, y_{t+1}, \; x_t \in \mathbb{Z}_+^{n_t}, \tag{4.21}$$

erweitert wird und diese Variablen mit obigen Strafkosten in die Zielfunktion eingehen. In (4.21) ist M eine hinreichend große Zahl und $\mathbf{1}$ ein Vektor dessen Einträge nur aus Einsen bestehen. Anders als im vorangegangenen Anwendungsbeispiel, haben wir hier bei einem entsprechend großen M den Fall der vollständigen Kompensation, vergleiche Kapitel 1.

In die Zielfunktion gehen also neben den positiven Vergütungen c_t, die negativen Strafkosten d_{t+1} ein und wir erhalten:

$$\sum_{t=1}^{T-1} c_t^\top x_t + d_{t+1}^\top \, y_{t+1} \longrightarrow \max ! \tag{4.22}$$

4.2.2 Testrechnungen

Wir wollen in diesem Abschnitt wiederum beide Algorithmen mit CPLEX vergleichen. Zusätzlich werden wir einen internen Vergleich zwischen bb-tree und bb-hash anstellen, der uns Aufschluss über die Anzahl berechneter Szenarioteilprobleme gibt. Die benutzte Hardware ist bei allen Testproblemen ein Sun Ultra Sparc III Rechner mit einem 1.2 GHz Prozessor und 32 GB RAM Arbeitsspeicher.

Zuerst werden wir uns dem Vergleich mit CPLEX widmen. Bei den Berechnungen wurde die CPLEX-Version 8.1 benutzt. In den Tabellen 4.12, 4.13 und 4.14 wurden Mean-Risk Modelle mit den ausgewiesenen Risikomaßen berechnet. Die erste Spalte dient der Identifikation der Probleme. Die nächsten vier Spalten spiegeln die Größe der berechneten Probleme wieder. Wie in Kapitel 4.1.3, wird durch das β die Gewichtung des Risikomaßes festgelegt.

Bei allen Testproblemen wurde eine Zeitschranke von zwei Stunden gewählt. Die Berechnung wurde ebenfalls abgebrochen, wenn ein relatives Gap, definiert wie in (4.17), von 0.1% innerhalb der zwei Stunden erreicht wurde.

Die letzten drei Spalten dokumentieren die Ergebnisse der jeweiligen Algorithmen. Die Einträge geben das relative Gap an, das in zwei Stunden berechnet wurde. Der Eintrag "-" steht für Rechnungen, bei denen das Gap innerhalb der Zeitschranke nicht unter 1000% fiel.

Wir sehen, dass sowohl bb-tree als auch bb-hash in allen Instanzen bessere Resultate lieferten als CPLEX. Vergleicht man nur die Ergebnisse von bb-tree und bb-hash, so ist eine Abhängigkeit der Resultate von den verschiedenen Risikomaßen erkennbar. So sind die Resultate von bb-hash bei der Excess Probability und beim Expected Excess besser, wogegen beim Conditional Value-at-Risk bb-tree bessere Ergebnisse liefert. Die Ursache liegt nach Meinung des Autors an zusätzlichen stetigen Variablen, die zur Induzierung des Conditional Value-at-Risk nötig sind. Diese erzeugen durch die individuelle Behandlung der Masterknoten in bb-hash einen erheblichen Aufwand, da es nur selten vorkommt, dass sich die Branchingentscheidungen wiederholen. Somit ist die mehrfache Nutzung bereits berechneter Szenarioprobleme eingeschränkt. Da bei bb-tree eine Branchingentscheidung auf jeder Ebene des

Test	Stufen	Szenarien	Variablen Stet/Ganz/Bin	Nebenbe-dingungen	Gewicht β	bb-tree	CPLEX	bb-hash
EP1			0/21/1080	1083	0	0.1%	468.7%	0.1%
EP2	3	48	/21/1138	1137	0.5	0.1%	699.3 %	0.1%
EP3			0/21/1138	1137	1	0.1%	–	0.1%
EP4			0/21/1138	1137	2	0.1%	–	0.1%
EP5			0/9/2000	2003	0	0.1%	271.1%	0.1%
EP6	3	96	0/9/2100	2103	0.5	0.1%	200.3%	0.1%
EP7			0/9/2100	2103	1	0.1%	532.1%	0.1%
EP8			0/9/2100	2103	2	0.1%	957.2%	0.1%
EP9			0/33/1160	1163	0	0.1%	344.4%	0.1%
EP10	4	48	0/33/1218	1221	0.5	0.1%	–	0.1%
EP11			0/33/1218	1221	1	60.4%	–	0.1%
EP12			0/33/1218	1221	2	0.1%	389.9%	0.1%
EP13			0/33/2120	2123	0	0.1%	567.6%	0.1%
EP14	4	96	0/33/2226	2229	0.5	43.8%	664.2%	0.1%
EP15			0/33/2226	2229	1	0.1%	–	0.1%
EP16			0/33/2226	2229	2	0.1%	–	0.1%
EP17			0/45/1178	1243	0	0.1%	300.5%	0.1%
EP18	5	48	0/45/1302	1305	0.5	148.7%	–	0.1%
EP19			0/45/1302	1305	1	80.3%	–	0.1%
EP20			0/45/1302	1305	2	33.8%	–	0.1%
EP21			0/45/2200	2203	0	0.1%	589.3%	0.1%
EP22	5	96	0/45/2310	2313	0.5	107.3%	–	0.1%
EP23			0/45/2310	2313	1	91.6%	–	0.1%
EP24			0/45/2310	2313	2	29.6%	–	0.1%
EP25			0/93/1560	1563	0	19.8%	570.3%	27.7%
EP26	6	48	0/93/1638	1641	0.5	34.8%	–	6.5%
EP27			0/93/1638	1641	1	18.0%	63.5%	0.1%
EP28			0/93/1638	1641	2	12.7%	46.6%	0.1%
EP29			0/93/2520	2523	0	10.1%	406.9%	41.9%
EP30	6	96	0/93/2646	2649	0.5	16.6%	–	33.6%
EP31			0/93/2646	2649	1	0.1%	72.5%	10.3%
EP32			0/93/2646	2649	2	2.7%	44.6%	4.2%
EP33			0/189/2310	2203	0	0.1%	211.8%	0.1%
EP34	7	48	0/189/2310	2313	0.5	31.9%	–	0.8%
EP35			0/189/2310	2313	1	14.8%	51.6%	1.3%
EP36			0/189/2310	2313	2	8.9%	30.9%	0.8%
EP37			0/189/3160	3163	0	83.9%	480.3%	31.2%
EP38	7	96	0/189/3318	3321	0.5	34.3%	–	24.2%
EP39			0/189/3318	3321	1	16.9%	53.4%	12.8%
EP40			0/189/3318	3321	2	11.7%	31.2%	12.0%

Tabelle 4.12: Exess Probability - Vergleich mit CPLEX

Test	Stufen	Szenarien	Variablen Stet/Ganz/Bin	Nebenbe-dingungen	Gewicht β	bb-tree	CPLEX	bb-hash
EE1			0/21/1080	1083	0	0.1%	468.7%	0.1%
EE2	3	48	54/21/1080	1137	0.5	0.1%	–	0.1%
EE3			54/21/1080	1137	1	0.1%	–	0.1%
EE4			54/21/1080	1137	2	0.1%	–	0.1%
EE5			0/9/2000	2003	0	0.1%	271.1%	0.1%
EE6	3	96	100/9/2000	2103	0.5	0.1%	991.1%	0.1%
EE7			100/9/2000	2103	1	0.1%	–	0.1%
EE8			100/9/2000	2103	2	0.1%	271.4%	0.1%
EE9			0/33/1160	1163	0	0.1%	344.4%	0.1%
EE10	4	48	58/33/1160	1221	0.5	11.3%	–	0.1%
EE11			58/33/1160	1221	1	9.9%	67.5%	0.1%
EE12			58/33/1160	1221	2	8.6%	57.8%	0.1%
EE13			0/33/2120	2123	0	0.1%	567.6%	0.1%
EE14	4	96	106/33/2120	2229	0.5	9.9%	26.3%	0.1%
EE15			106/33/2120	2229	1	19.1%	79.6%	0.1%
EE16			106/33/2120	2229	2	15.9%	81.9%	0.1%
EE17			0/45/2200	1243	0	0.1%	300.5%	0.1%
EE18	5	48	62/45/1240	1305	0.5	44.0%	–	0.1%
EE19			62/45/1240	1305	1	21.1%	–	0.1%
EE20			62/45/1240	1305	2	14.1%	–	0.1%
EE21			0/45/2200	2203	0	0.1%	589.3%	0.1%
EE22	5	96	110/45/2200	2313	0.5	44.9%	–	0.1%
EE23			110/45/2200	2313	1	18.5%	83.7%	0.1%
EE24			110/45/2200	2313	2	12.0%	58.3 %	0.1%
EE25			0/93/1560	1563	0	19.8%	570.3%	27.7%
EE26	6	48	78/93/1560	1641	0.5	66.9%	–	21.9%
EE27			78/93/1560	1641	1	32.6%	63.5%	4.8%
EE28			78/93/1560	1641	2	24.8%	46.6%	1.5%
EE29			0/93/2520	2523	0	10.1%	406.9%	41.9%
EE30	6	96	126/93/2520	2649	0.5	4.8%	–	0.1%
EE31			126/93/2520	2649	1	0.1%	57.8%	0.1%
EE32			126/93/2520	2649	2	0.1%	14.3%	0.1%
EE33			0/189/2200	2203	0	0.1%	211.8%	0.1%
EE34	7	48	110/189/2200	2313	0.5	35.8%	–	0.1%
EE35			110/189/2200	2313	1	21.7%	57.2%	0.1%
EE36			110/189/2200	2313	2	14.9%	36.6%	0.1%
EE37			0/189/3160	3163	0	83.9%	480.3%	31.2%
EE38	7	96	158/189/3160	3321	0.5	80.9%	–	38.8%
EE39			158/189/3160	3321	1	24.3%	60.9%	0.1%
EE40			158/189/3160	3321	2	16.8%	42.9%	0.1%

Tabelle 4.13: Expected Exess - Vergleich mit CPLEX

Test	Stufen	Szenarien	Variablen Stet/Ganz/Bin	Nebenbe-dingungen	Gewicht β	bb-tree	CPLEX	bb-hash
CV1			0/21/1080	1083	0	0.1%	468.7%	0.1%
CV2	3	48	56/21/1080	1137	0.5	0.1%	–	0.1%
CV3			56/21/1080	1137	1	0.1%	–	0.1%
CV4			56/21/1080	1137	2	0.1%	–	0.1%
CV5			0/9/2000	2003	0	0.1%	271.1%	0.1%
CV6	3	96	102/9/2000	2103	0.5	0.1%	520.1%	0.1%
CV7			102/9/2000	2103	1	0.1%	359.7%	0.1%
CV8			102/9/2000	2103	2	0.1%	–	0.1%
CV9			0/33/1160	1163	0	0.1%	344.4%	0.1%
CV10	4	48	61/33/1160	1221	0.5	0.1%	430.5%	0.1%
CV11			61/33/1160	1221	1	0.1%	377.1%	0.1%
CV12			61/33/1160	1221	2	1,3%	–	21.7%
CV13			0/33/2120	2123	0	0.1%	567.6%	0.1%
CV14	4	96	109/33/2120	2229	0.5	0.1%	–	0.1%
CV15			109/33/2120	2229	1	0.1%	–	0.1%
CV16			109/33/2120	2229	2	2,3%	–	55.3%
CV17			0/45/2200	1243	0	0.1%	300.5%	0.1%
CV18	5	48	66/45/1240	1305	0.5	0.1%	443.2%	0.1%
CV19			66/45/1240	1305	1	0.1%	521.8%	1.7%
CV20			66/45/1240	1305	2	99.1%	–	71.1%
CV21			0/45/2200	2203	0	0.1%	589.3%	0.1%
CV22	5	96	114/45/2200	2313	0.5	0.2%	–	0.2%
CV23			114/45/2200	2313	1	2.1%	–	4.9%
CV24			114/45/2200	2313	2	15.1%	904.5%	23.1%
CV25			0/93/1560	1563	0	19.8%	570.3%	27.7%
CV26	6	48	83/93/1560	1641	0.5	29.6%	–	33.9%
CV27			83/93/1560	1641	1	35.3%	–	34.6%
CV28			83/93/1560	1641	2	33.2%	–	51.7%
CV29			0/93/2520	2523	0	10.1%	406.9%	41.9%
CV30	6	96	131/93/2520	2649	0.5	19.5%	561.4%	35.6%
CV31			131/93/2520	2649	1	35.4%	476.7%	42.3%
CV32			131/93/2520	2649	2	46.1%	–	60.5%
CV33			0/189/2200	2203	0	0.1%	211.8%	0.1%
CV34	7	48	116/189/2200	2313	0.5	34.9%	341,1%	35.3%
CV35			116/189/2200	2313	1	36,4%	453.2%	37.7%
CV36			116/189/2200	2313	2	35.5%	167.9%	42.2%
CV37			0/189/3160	3163	0	83.9%	480.3%	31.2%
CV38	7	96	164/189/3160	3321	0.5	61.8%	–	34.1%
CV39			164/189/3160	3321	1	3.9%	902.1%	43.6%
CV40			164/189/3160	3321	2	6.2%	–	54.0%

Tabelle 4.14: Conditional Value-at-Risk - Vergleich mit CPLEX

Masterbaumes festgelegt wird, ist eine mehrfache Nutzung von berechneten Szenarioproblemen entsprechend häufiger. Zwar kommt es beim Expected Excess auch zu zusätzlichen stetigen Variablen, die Nichtantizipativitätsbedingungen sind jedoch erfahrungsgemäß einfacher zu "reparieren".

Wir kommen nun zu dem angekündigten internen Vergleich zwischen bb-tree und bb-hash. Von zentraler Bedeutung ist der Einspareffekt, der durch die mehrfache Nutzung bereits berechneter Szenarioteilprobleme erzielt wird. In den Kapiteln 3.3.1 und 3.3.2 haben wir gesehen, dass schon bei einem sehr kleinen Problem ein Drittel weniger Szenarioprobleme berechnet werden musste als in der Basisversion in Kapitel 3.3.

In den Tabellen 4.15, 4.16 und 4.17 sind entsprechende Daten für die oben vorgestellten Testprobleme zusammengetragen. In den Spalten drei und vier wird dokumentiert, wieviel Master- bzw. Szenarioknoten in bb-tree bei Abbruch erzeugt wurden. Die entsprechenden Informationen sind für bb-hash in den Spalten sieben und acht zu sehen. Die Zahlen sind Vielfache von eintausend bzw. von einer Million. In der fünften bzw. neunten Spalte steht die relative Anzahl berechneter Szenarioknoten, im Vergleich zur Basisversion in Kapitel 3.3. Das heißt, 100% entspricht der Anzahl an zu berechnenden Szenarioteilproblemen, wenn kein Verwaltungssystem benutzt worden wäre. In den Spalten zwei und sechs sind noch einmal die erreichten Gaps nach obigen Abbruchkriterien zu sehen. Um einen Eindruck davon zu bekommen, welche Ergebnisse ohne Verwaltungssystem zu erzielen wären, stehen in der letzten Spalte die berechneten Gaps der Basisversion. Hier steht der Eintrag "-" wiederum für ein Gap, welches größer als 1000% war.

Wir sehen, dass bei vielen Testproblemen weniger als 1% -verglichen mit der Basisversion- der Szenarioteilprobleme berechnet wurden. Wie die absoluten Zahlen veranschaulichen, wurden teilweise weit mehr Master- als Szenarioknoten erzeugt, siehe etwa die Testergebnisse von EP9 bis EP40 bei bb-tree oder EP25 bis EP28 bei bb-hash. Das bedeutet, dass im Durchschnitt weniger als ein Szenarioteilproblem berechnet werden musste, um die Informationen eines Masterproblems mit der entsprechenden Szenarioanzahl zu erhalten. Die letzte Spalte der Tabellen 4.15 - 4.17 dokumentieren, dass die Ergebnisse ohne Verwaltungssystem zum größten Teil weit

Test	bb-tree	M-Knoten	S-Knoten	Berechnete Probleme	bb-hash	M-Knoten	S-Knoten	Berechnete Probleme	Basis-version
EP1	0.1%	4.6T	2.5T	1.142%	0.1%	5.0T	2.6T	1.086%	0.1%
EP2	0.1%	104.0T	5.4T	0.108%	0.1%	4.3T	2.5T	1.248%	59.2%
EP3	0.1%	183.0T	7.7T	0.087%	0.1%	3.0T	2.5T	1.766%	248.0%
EP4	0.1%	292.0T	8.8T	0.063%	0.1%	2.0T	2.4T	2.566%	54.4%
EP5	0.1%	0.1T	2.8T	68.604%	0.1%	0.1T	2.5T	67.853%	0.1%
EP6	0.1%	0.1T	2.3T	26.373%	0.1%	0.1T	1.9T	100.00%	0.1%
EP7	0.1%	0.1T	1.6T	70.000%	0.1%	0.1T	1.7T	100.00%	0.1%
EP8	0.1%	0.1T	1.0T	64.705%	0.1%	0.1T	1.1T	100.00%	0.1%
EP9	0.1%	7.5T	4.4T	1.210%	0.1%	10.1T	4.6T	0.950%	–
EP10	0.1%	10.1M	7.0T	0.001%	0.1%	7.0T	4.6T	1.371%	229.9%
EP11	60.4%	7.5M	5.8T	0.001%	0.1%	4.0T	4.6T	2.400%	145.1%
EP12	0.1%	23.0M	7.8T	0.001%	0.1%	4.0T	4.9T	2.581%	64.7%
EP13	0.1%	2.3T	6.2T	2.876%	0.1%	2.1T	6.7T	3.345%	–
EP14	43.8%	3.3M	10.3T	0.003%	0.1%	2.0T	6.2T	3.227%	198.7%
EP15	0.1%	8.7M	14.4T	0.002%	0.1%	1.0T	5.0T	6.075%	116.9%
EP16	0.1%	8.9M	15.1T	0.002%	0.1%	1.0T	6.1T	6.400%	58.2%
EP17	0.1%	1.0M	44.9T	0.093%	0.1%	1.4M	58.3T	0.085%	511.1%
EP18	148.7%	50.2M	47.1T	0.003%	0.1%	1.6M	59.0T	0.076%	–
EP19	80.3%	48.8M	41.2T	0.002%	0.1%	2.2M	59.4T	0.056%	144.6%
EP20	33.8%	47.2M	38.0T	0.002%	0.1%	2.3M	58.4T	0.052%	71.0%
EP21	0.1%	600.0T	83.5T	0.145%	0.1%	519.7T	89.3T	0.179%	656.8%
EP22	107.3%	20.4M	84.6T	0.193%	0.1%	470.0T	87.0T	0.192%	–
EP23	91.6%	19.2M	70.1T	0.004%	0.1%	430.0T	82.9T	0.201%	172.8%
EP24	29.6%	20.7M	70.4T	0.004%	0.1%	390.0T	76.8T	0.205%	82.3%
EP25	19.8%	73.1M	50.3T	0.006%	27.7%	39.4M	32.6T	0.002%	394.4%
EP26	34.8%	57.4M	13.3T	0.002%	6.5%	16.4M	30.3T	0.004%	56.4%
EP27	18.0%	54.4M	13.5T	0.001%	0.1%	19.2M	40.0T	0.004%	31.6%
EP28	12.7%	72.3M	16.8T	0.002%	0.1%	2.8M	28.8T	0.021%	24.3%
EP29	10.1%	32.3M	60.9T	0.005%	41.9%	733.0T	29.0T	0.041%	918.2%
EP30	16.6%	41.3M	25.7T	0.008%	33.7%	1.0T	5.4T	5.725%	33.7%
EP31	0.1 %	512.0T	8.8T	0.018%	10.3%	10.0T	6.9T	0.725%	31.5%
EP32	2.7 %	995.0T	11.7T	0.012%	4.2 %	1.0T	4.9T	5.200%	29.2%
EP33	0.1%	124.0T	9.9T	0.167%	0.1%	36.0T	9.4T	0.544%	896.6%
EP34	31.9%	6.4M	4.9T	0.001%	0.8%	665.0T	20.2T	0.047%	55.8%
EP35	14.8%	5.1M	4.9T	0.002%	1.3%	175.0T	15.9T	0.142%	29.9%
EP36	8.9 %	14.3M	5.3T	0.001%	0.8%	152.0T	14.6T	0.150%	16.7%
EP37	83.9%	47.9T	26.6T	0.579%	31.2%	29.0T	22.8T	0.822%	–
EP38	34.3%	39.7M	23.1T	0.004%	24.2%	1.1T	6.8T	6.454%	59.9%
EP39	16.9%	23.6M	19.8T	0.001%	12.8%	0.2T	3.7T	19.500%	33.1%
EP40	11.7%	2.8M	15.3T	0.005%	12.0%	0.1T	2.7T	29.000%	23.6%

Tabelle 4.15: Exess Probability - Vergleich intern

Test	bb-tree	M-Knoten	S-Knoten	Berechnete Probleme	bb-hash	M-Knoten	S-Knoten	Berechnete Probleme	Basis-version
EE1	0.1%	4.6T	2.5T	1.142%	0.1%	5.0T	2.6T	1.086%	0.1%
EE2	0.1%	359.0T	4.5T	0.026%	0.1%	12.0T	2.0T	0.353%	142.8%
EE3	0.1%	314.8T	4.3T	0.029%	0.1%	21.7T	2.2T	0.204%	146.0%
EE4	0.1%	111.2T	3.1T	0.058%	0.1%	13.8T	2.1T	0.319%	206.4%
EE5	0.1%	0.1T	2.8T	68.604%	0.1%	0.1T	2.5T	67.853%	0.1 %
EE6	0.1%	0.1T	3.4T	9.000%	0.1%	0.1T	1.7T	100.000%	0.1 %
EE7	0.1%	0.1T	2.6T	22.689%	0.1%	0.1T	1.5T	100.000%	0.1%
EE8	0.1%	0.5T	4.5T	9.5 %	0.1%	0.1T	2.7T	58.000%	0.1 %
EE9	0.1%	7.5T	4.4T	1.210%	0.1%	10.1T	4.6T	0.950%	–
EE10	11.3%	12.3M	5.5T	0.001%	0.1%	0.3T	3.5T	2.490%	81.4%
EE11	9.9%	11.3M	4.5T	0.001%	0.1%	2.0T	3.0T	3.188%	51.4%
EE12	8.6%	12.7M	4.2T	0.001%	0.1%	3.0T	3.2T	2.241%	46.9%
EE13	0.1%	2.3T	6.2T	2.876%	0.1%	2.1T	6.7T	3.345%	–
EE14	9.9%	55.6M	8.1T	0.001%	0.1%	4.9T	5.0T	1.059%	18.4%
EE15	19.1%	26.7M	11.5T	0.001%	0.1%	2.2T	5.4T	2.591%	40.4%
EE16	15.9%	53.9M	11.5T	0.001%	0.1%	6.0T	5.7T	0.984%	41.6%
EE17	0.1%	1.0M	44.9T	0.093%	0.1%	1.4M	58.3T	0.085%	511.1%
EE18	44.0%	9.3M	28.5T	0.006%	0.1%	625.0T	42.0T	0.140%	76.3%
EE19	21.1%	9.6M	25.1T	0.005%	0.1%	376.0T	32.9T	0.182%	36.7%
EE20	14.1%	9.7M	25.5T	0.006%	0.1%	262.0T	28.1T	0.223%	25.8%
EE21	0.1%	600.0T	83.5T	0.145%	0.1%	519.7T	89.3T	0.179%	656.8%
EE22	44.9%	4.2M	53.7T	0.013%	0.1%	325.0T	67.1T	0.215%	82.6%
EE23	18.5%	4.7M	37.2T	0.008%	0.1%	241.0T	59.0T	0.255%	43.3%
EE24	12.0%	4.5M	36.8T	0.008%	0.1%	196.0T	56.7T	0.301%	31.3%
EE25	19.8%	73.1M	50.3T	0.006%	27.7%	39.4M	32.6T	0.002%	394.4%
EE26	66.9%	7.5M	7.3T	0.002%	21.9%	5.5M	16.9T	0.006%	91.29%
EE27	32.6%	7.7M	6.6T	0.002%	4.8%	5.8M	17.3T	0.006%	52.2%
EE28	24.8%	7.6M	6.2T	0.002%	1.5%	5.8M	18.1T	0.006%	39.6%
EE29	10.1%	32.3M	60.9T	0.005%	41.9%	733.0T	29.0T	0.041%	918.2%
EE30	4.8%	4.1M	15.6T	0.004%	0.1%	45.0T	9.0T	0.208%	32.7%
EE31	0.1%	0.1T	0.1T	80.000%	0.1%	0.1T	0.1T	80.000%	0.1%
EE32	0.1%	0.1T	0.1T	100.000%	0.1%	0.1T	0.1T	100.000%	0.1%
EE33	0.1%	124.0T	9.9T	0.167%	0.1%	36.0T	9.4T	0.544%	896.6%
EE34	35.8%	3.1M	4.4T	0.002%	0.1%	1.0M	20.2T	0.031%	58.6%
EE35	21.7%	3.6M	4.1T	0.002%	0.1%	44.0T	12.9T	0.467%	27.0%
EE36	14.9%	3.5T	3.9T	0.002%	0.1%	11.0T	8.5T	1.211%	18.7%
EE37	83.9%	47.9T	26.6T	0.579%	31.2%	29.0T	22.8T	0.822%	–
EE38	80.9%	3.2M	17.8T	0.005%	38.8%	66.0T	28.5T	0.450%	104.7%
EE39	24.3%	3.3M	17.3T	0.005%	0.1%	30.0T	22.0T	0.765%	36.1%
EE40	16.8%	3.5M	14.6T	0.004%	0.1%	13.0T	16.4T	1.320%	25.6%

Tabelle 4.16: Expected Exess - Vergleich intern

Test	bb-tree	M-Knoten	S-Knoten	Berechnete Probleme	bb-hash	M-Knoten	S-Knoten	Berechnete Probleme	Basis-version
CV1	0.1%	4.6T	2.5T	1.142%	0.1%	5.0T	2.6T	1.086%	0.1%
CV2	0.1%	1.3M	15.6T	0.025%	0.1%	1.1M	97.9T	0.183%	122.9%
CV3	0.1%	7.2M	17.1T	0.005%	0.1%	7.3M	275.4T	0.079%	962.1%
CV4	0.1%	12.3M	13.2T	0.002%	0.1%	14.0M	370.8T	0.055%	–
CV5	0.1%	0.1T	2.8T	68.604%	0.1%	0.1T	2.5T	67.853%	0.1%
CV6	0.1%	3.0T	61.7T	21.433%	0.1%	6.0T	154.8T	26.87%	0.1%
CV7	0.1%	1.0T	23.4T	21.602%	0.1%	91.6T	178.8T	2.033%	55.4%
CV8	0.1%	3.4T	33.2T	10.073%	0.1%	1.5M	295.6T	0.197%	–
CV9	0.1%	7.5T	4.4T	1.210%	0.1%	10.1T	4.6T	0.950%	–
CV10	0.1%	176.1T	83.5T	0.502%	0.1%	157.1T	373.0T	4.946%	–
CV11	0.1%	2.2M	152.5T	0.144%	0.1%	1.3M	976.1T	1.576%	–
CV12	1,3%	6.4M	119.1T	0.36%	21.7%	6.5M	743.9T	0.236%	–
CV13	0.1%	2.3T	6.2T	2.876%	0.1%	2.1T	6.7T	3.345%	–
CV14	0.1%	76.9T	179.6T	2.434%	0.1%	80.6T	1.0M	13.522%	50.3%
CV15	0.1%	17.3T	13.9T	0.439%	0.1%	406.5T	1.0M	2.592%	53.8%
CV16	2.3%	50.8M	169.4T	0.029%	55.3%	2.6M	895.0T	0.360%	60.8%
CV17	0.1%	1.0M	44.9T	0.093%	0.1%	1.4M	58.3T	0.085%	511.1%
CV18	0.1%	10.1M	291.1T	0.060%	0.1%	4.1M	54.2T	0.272%	617.5%
CV19	0.1%	40.5M	640.6T	0.033%	1.7%	9.1M	459.9T	0.105%	716.1%
CV20	99.1%	44.5M	160.7T	0.008%	71.1%	3.2M	139.8T	0.009%	818.3%
CV21	0.1%	600.0T	83.5T	0.145%	0.1%	519.7T	89.3T	0.179%	656.8%
CV22	0.2%	3.7M	933.2T	0.263%	0.2%	1.1M	321.7T	0.299%	750.6%
CV23	2.1%	9.6M	841.1T	0.091%	4.9%	2.1M	294.4T	0.143%	823.2%
CV24	15.1%	22.2M	525.5T	0.024%	23.1%	2.2M	116.8T	0.055%	–
CV25	19.8%	73.1M	50.3T	0.006%	27.7%	39.4M	32.6T	0.002%	394.4%
CV26	29.6%	42.1M	40.6T	0.002%	33.9%	20.4T	48.1T	0.005%	454.1%
CV27	35.3%	79.4M	43.6T	0.009%	34.6%	20.1T	47.5T	0.005%	462.2%
CV28	33.2%	85.7M	44.3T	0.024%	51.7%	20.6T	47.5T	0.005%	620.8%
CV29	10.1%	32.3M	60.9T	0.005%	41.9%	733.0T	29.0T	0.041%	918.2%
CV30	19.5%	31.7M	53.1T	0.004%	35.6%	3.4M	34.6T	0.011%	–
CV31	35.4%	35.3M	55.4T	0.006%	42.3%	4.0M	33.4T	0.009%	615.4%
CV32	46.1%	22.4M	45.9T	0.002%	60.5%	4.7M	33.5T	0.007%	–
CV33	0.1%	124.0T	9.9T	0.167%	0.1%	36.0T	9.4T	0.544%	896.6%
CV34	34.9%	310.0T	131.4T	0.883%	35.3%	66.0T	111.1T	3.508%	279.8%
CV35	36.4%	1.1M	202.1T	0.376%	37.7%	105.0T	124.4T	2.469%	279.0%
CV36	35.5%	8.9M	265.3T	0.062%	42.2%	362.0T	135.2T	0.777%	336.6%
CV37	83.9%	47.9T	26.6T	0.579%	31.2%	29.0T	22.8T	0.822%	–
CV38	61.8%	110.0T	34.7T	0.329%	34.1%	25.0T	21.3T	0.891%	–
CV39	3.9%	749.0T	60.3T	0.043%	43.6%	17.0T	18.9T	1.162%	–
CV40	6.2%	253.7T	14.5T	0.184%	54.0%	48.0T	27.9T	0.607%	–

Tabelle 4.17: Conditional Value-at-Risk - Vergleich intern

schlechter ausfallen.

	Excess Probability	Expected Excess	Conditional Value-at-Risk
bb-tree	0.13%	0.18%	0.89%
bb-hash	1.28%	1.20%	7.78%

Tabelle 4.18: Durchschnittlicher Anteil an berechneten Szenarioknoten

Vergleicht man bb-tree und bb-hash, so lässt sich feststellen, dass die Anzahl erzeugter Masterknoten bei bb-hash im Durchschnitt geringer ist, was an der individuellen Behandlung der Masterknoten liegt. Diese individuelle Behandlung führt auch zu einem prozentual höheren Anteil erzeugter Szenarioknoten. In Tabelle 4.18 wird dies veranschaulicht. Es wurden die in allen Testrechnungen berechneten Szenarioknoten ins Verhältnis zu den insgesamt berechneten Masterknoten gesetzt. Man sieht, dass der prozentuale Anteil berechneter Szenarioknoten bei bb-hash um ungefähr das zehnfache höher ist.

Kapitel 5

Zusammenfassung

Bei vielen Entscheidungsproblemen ist die Berücksichtigung zufallsabhängiger Daten unverzichtbar. Die stochastische Optimierung stellt den Entscheidungsträgern Werkzeuge zur Verfügung, mit denen solche Probleme mathematisch formuliert und berechnet werden können. Dazu werden verschiedene Modelle bereitgestellt.

In dieser Arbeit betrachteten wir das mehrstufige Kompensationsmodell für lineare, gemischt-ganzzahlige Probleme. Wir haben das klassische erwartungswertbasierte Modell mithilfe von Risikomaßen zu einem Mean-Risk Modell erweitert. Bei der Wahl der Risikomaße stand der algorithmische Aspekt im Mittelpunkt, der in dieser Arbeit den Schwerpunkt setzte.

Für den Fall einer diskreten Wahrscheinlichkeitsverteilung wurde ein Dekompositionsverfahren vorgestellt, das die Berechnung von Probleminstanzen erlaubt, die aufgrund ihrer Größe von Standardlösern nur eingeschränkt bearbeitet werden können. Aus diesem Dekompositionsverfahren gingen die konkreten Algorithmen bb-tree und bb-hash hervor, welche sich durch unterschiedliche Speicherverwaltungssysteme unterscheiden.

Es wurden zwei Anwendungen vorgestellt, mit deren Hilfe die konzeptionellen Unterschiede verschiedener Mean-Risk Modelle veranschaulicht wurden. Des Weiteren wurden anhand von Testrechnungen die Rechengeschwindigkeiten der Algorithmen mit dem kommerziell verfügbaren Programm CPLEX verglichen. Die Testrechnungen haben gezeigt, dass bb-tree und bb-hash als Alternative zu Standardlösern wie beispielsweise CPLEX in Betracht gezogen werden können. Ein interner Ver-

gleich zwischen den beiden Speicherverwaltungssystemen, der Aufschluss auf das Rechenverhalten der Algorithmen gab, beendete diese Arbeit.

Literaturverzeichnis

[1] Acerbi, C., Tasche, D., *On the coherence of Expected Shortfall*, Journal of Banking and Finance 26, 2002, pp. 1487-1503

[2] Ackermann, T., Andersson, G., Söder, L., *Distributed Generation: a Definition*, Electric Power System Research 57, 2001

[3] Ahmed, S., King, A.J., Parija, G., *A Multi-stage Stochastic Integer Programming Approach for Capacity Expansion under Uncertainty* , Journal of Global Optimization 26, 2003, pp. 3-24

[4] Albers, S., *Online algorithms: A survey*, Mathematical Programming 97, 2003, pp. 3-26

[5] Alonso-Ayuso, A., Escudero, L.F., Garín, A., Ortuño, M.T., Pérez, G., *An approach for strategic supply chain planning under uncertainty based on stochastic 0-1 programming*, Journal of Global Optimization 26, 2003, pp. 97-124

[6] Alonso-Ayuso, A., Escudero, L.F., Ortuño, M.T., *BFC, A branch-and-fix coordination algorithmic framework for solving some types of stochastic pure and mixed 0-1 programs*, European Journal of Operations Research 151, 2003, pp. 503-519

[7] Andersson, G., *Modelling and Analysis of Electric Power Systems*, Lecture 227-0526-00, Power Systems Laboratory, ETH Zürich, 2004

[8] Artzner, P., Delbaen, F., Eber, J.-M., Heath, D., *Coherent measures of risk* , Mathematical Finance 9, 1999, pp. 203-228

[9] Artzner, P., Delbaen, F., Eber, J.-M., Heath, D., Ku, H., *Coherent multiperiod Risk Measurement*, Working Paper, www.math.rthz.ch/ delbaen, 2002

[10] Artzner, P., Delbaen, F., Eber, J.-M., Heath, D., Ku, H., *Coherent multiperiod risk adjusted values and Bellman's principle*, Preprint, Department of Mathematics, ETH Zürich, 2004

[11] Bauer, H., *Wahrscheinlichkeitstheorie*, de Gryter, 2002

[12] Beale, E.M.L., *On minimizing a convex function subject to linear inequalities*, J. Royal Statistical Society, Series B 17, 1955, pp. 173-184

[13] Benders, J.F., *Partitioning procedures for solving mixed-variables programming problems*, Numerische Mathematik 4, 1962, pp. 238-252

[14] Bereanu, B., *Stable stochastic linear programs and applications*, Mathematische Operationsforschung und Statistik 6, 1975, pp. 593-607

[15] Ben-Tal, A., Ghaoui, E., Nemirovski, A., *Robust Semidefinite Programming*, In: Wolkowitcz, H., Saigal, R., Vandenberghe, L., eds., Handbook on Semidefinite Programming, Kluwer Academic Publishers, 2000, pp. 139-162

[16] Ben-Tal, A., Nemirovski, A., *Robust Optimization - methodology and applications* , Mathematical Programming 92, Series B, pp.453-480, 2002

[17] Birge, J.R., Long, E., Takriti, S., *A stochastic model for the unit commitment problem*, IEEE Transactions on Power Systems 11, pp. 1497-1508, 1996

[18] Birge, J.R., Louveaux, F., *Introduction to Stochastic Programming*, Springer, New York, 1997

[19] Broy, M., *Informatik: Eine grundlegende Einführung, Band2*, Springer, 1998

[20] Carøe, C.C.; Schultz, R., *Dual Decomposition in Stochastic Integer Programming*, Operations Research Letters 24, 1999, pp. 37-45

[21] Charnes, A., Cooper, W.W., *Cost horizons and certainty equivalents: an approach to stochastic programming of heating oil*, Management Science 6, 1958, pp. 235-263

[22] Charnes, A., Cooper, W.W., *Chance-constrained programming*, Management Science 6, 1959, pp. 73-79

[23] Charnes, A., Cooper, W.W., *Deterministic equivalents for optimizing and satisficing under chance constraints*, Operations Research 11, 1963, pp. 18-39

[24] Chvátal, V., *Linear Programming*, Freeman, 1983

[25] Czech, Z.J., Havas, G., Majewski, B.S., *An optimal algorithm for generating minimal perfect hash functions*, Information Processing Letters 43, 1992, pp. 257-264

[26] Dantzig, G.B., *Maximization of a Linear Function of Variables Subject to Linear Inequalities*, T.C. Koopmans (Ed.), Activity Analysis of Production and Allocation, Wiley, New York, 1951, pp. 339-347

[27] Dantzig, G.B., *Linear Programming under Uncertainty*, Management Science 1, 1955, pp. 197-206

[28] Dash Optimization Xpress-MP, *Applications of optimization with Xpress-MP*, Dash Optimization Ltd., www.dashoptimization.com/home/downloads/book/booka4.pdf, 2000

[29] Dentcheva, D., Römisch, W., *Duality Gaps in nonconvex stochastic optimization*, Mathematical Programming 101, 2004, pp. 515-535

[30] Dentcheva, D., Ruszczyński, A., *Optimization with stochastic dominance constraints*, SIAM Journal on Optimization 14, 2003, pp. 548-566

[31] Dentcheva, D., Ruszczyński, A., *Portfolio Optimization with Stochastic Dominance Constraints*, Journal of Banking and Finance 30/2, 2006, pp. 433-451

[32] Dyer, M., Stougie, L., *Computational complexity of stochastic programming problems*, Mathematical Programming 106, 2006, pp. 423-432

[33] Ehrgott, M., *Multicriteria Optimization*, Springer, Berlin, 2000

[34] Eichhorn, A., Römisch, W., *Polyhedral risk measures in stochastic programming*, SIAM Journal on Optimization 16, 2005, pp. 69-95

[35] Ernst, H., *Grundlagen und Konzepte der Informatik*, Vieweg, 2002

[36] Fishburn, P.C., *Mean-Risk Analysis with Risk Associated with Below-Target Returns*, American Economic Review 67/2, 1977, pp. 116-126

[37] Fiedler, O., Römisch, W., *Stability in multistage stochastic programming*, Annals of Operations Research 56, 1995, pp. 79-93

[38] Föllmer, H., Schied, A., *Convex Risk measures and trading constraints*, Finance and Stochastic 6, 2002, pp. 429-447

[39] Gassmann, H.I., *The SMPS Format for Stochastic Linear Programs*, Applications of Stochastic Programming (Wallace, S.W., Ziemba, W.T., Eds.), MPS-SIAM Series in Optimization, 2005, pp. 9-19

[40] Gassmann, H.I., Schweitzer, E., *A comprehensive input format for stochastic linear programs*, Annals of Operations Research 104, 2001, pp. 89-125

[41] Gollmer, R., Gotzes, U., Schultz, R., *Stochastic Programs with Second-Order Dominance Constraints Induced by Mixed-Integer Linear Recourse*, Preprint 644-2007, Schriftreihen des FB Mathematik, Universität Duisburg-Essen, 2006

[42] Gollmer, R., Gotzes, U., Neise, F., Schultz, R., *Risk Modelling via Stochastic Dominance in Power Systems with Dispersed Generation*, Preprint 651-2006, Schriftreihen des FB Mathematik, Universität Duisburg-Essen, 2007

[43] Gollmer, R., Neise, F., Schultz, R., *Stochastic Programs with First-Order Dominance Constraints Induced by Mixed-Integer Linear Recourse*, Preprint 641-2006, Schriftreihen des FB Mathematik, Universität Duisburg-Essen, 2006

[44] Guan, Y., Ahmed, S., Nemhauser, G., *Cutting planes for multi-stage stochastic integer programs*, Stochastic Programming E-Print Series (SPEPS) 18, 2006

[45] Handschin, E., *Elektrische Energieübertragungssysteme*, 2. Auflage, Hüthig-Verlag, 1987

[46] Handschin, E., Neise, F., Neumann, H., Schultz, R., *Optimal operation of dispersed generation under uncertainty using mathematical programming*, presented at PSCC 2005 Liege/Belgium, International Journal of Power and Energy Systems (to appear)

[47] Havas, G., Majewski, B.S., *Optimal Algorithms for Minimal Perfect Hashing*, Technical Report 234, The University of Queensland, 1992

[48] Heinze, T., *User's guide to bb-tree - A C-Package for the Decomposition of Mixed-Integer Stochastic Programs, using Scenario Trees*, University of Duisburg-Essen, Department of Mathematics, 2005

[49] Heinze, T., *User's guide to bb-hash - A C-Package for the Decomposition of Mixed-Integer Stochastic Programs, using Hash Tables*, University of Duisburg-Essen, Department of Mathematics, 2006

[50] Heinze, T., Schultz, R., *A branch-and-bound method for multistage stochastic integer programs with risk objectives*, Optimization, 2007, (to appear)

[51] Heitsch, H., Römisch, W., *Scenario tree modelling for multistage stochastic programs*, Preprint 296, DFG Research Center Matheon Mathematics for key technologies", 2005

[52] Heitsch, H., Römisch, W., Strugarek, C., *Stability of multistage stochastic programs*, SIAM Journal on Optimization 17, 2006, pp. 511-525.

[53] Heitsch, H., Römisch, W., *A note on scenario reduction for two-stage stochastic programs*, Operations Research Letters, 2006, (to appear)

[54] Higle, J.L., Sen, S., *Stochastic Decomposition* , Kluwer Academic Publishers, 1996

[55] Hiriart-Urruty, J.-B., Lemaréchal, C., *Fundamentals of convex analysis*, Springer-Verlag, Berlin, 2001

[56] IEEE-Bus-Systems: http://www.ee.washington.edu/research/pstca/

[57] ILOG CPLEX, *User's Manual*, ILOG Inc., Mountain View, CA, 2002, www.cplex.com

[58] Jorion, P., *Value at Risk: The New Benchmark for Managing Financial Risk*, McGraw-Hill, 1997

[59] Kall, P., *Stochastic Linear Programming* , Springer, Berlin, 1976

[60] Kall, P., Mayer, J., *Stochastic Linear Programming*, Springer, 2005

[61] Kall, P., Wallace, S.W., *Stochastic Programming*, Wiley Chichester, 1994

[62] Kanková, V., *Stability in stochastic programming*, Kybernetika 14, 1978, pp. 339-349

[63] Kellerer, H., Pferschy, U., Pisinger, D., *Knapsack Problems*, Springer, 2004

[64] W.K. Klein Haneveld, M.H. van der Vlerk, *Stochastic integer programming: General models and algorithms* , Annals of Operations Research 85, 1999, pp. 39-57

[65] Korhonen, P., *Multiple Objective Programming Support, In: C.A. Floudas, P.M. Pardalos (Eds.), Encyclopedia of Optimization III*, Kluwer, Dordrecht, 2001, pp.566-574

[66] Kuhn, S., Schultz, R., *Risk Neutral and Risk Averse Power Optimization in Electricity Networks with Dispersed Generation*, Preprint 631-2006, Schriftreihe des FB Mathematik, Universität Duisburg-Essen, 2006

[67] Laporte, G., Louveaux, F.V., *The Integer L-Shaped Method for Stochastic Integer Programs with Complete Recourse*, Operations Research Letters 13, 1993, pp. 133-142

[68] Levy, H., *Stochastic Dominance and Expected Utility: Survey and Analysis*, Management Science 38, 1992, pp. 555-593

[69] Louveaux, F.V., Schultz, R., *Stochastic Integer Programming*, Stochastic Programming, Handbooks in Operations Research and Management Science Vol. 10 (A. Ruszczyński, A. Shapiro, Eds.), Elsevier, Amsterdam, 2003, pp.213-266

[70] Madansky, A, *Inequalities for Stochstic Linear Programming Problems*, Management Science 6, 1960, pp. 197-204

[71] Märkert A., *Deviation measures in stochastic programming with mixed-integer recourse*, Dissertation, Institute of Mathematics, University Duisburg-Essen, 2004

[72] Märkert A., Schultz, R., *On deviation measures in stochastic integer programming* , Operations Research Letters 33, 2005, pp. 441-449

[73] Markowitz, H.M., *Portfolio Selection: Efficient Diversification of Investments*, Wiley, New York, 1959

[74] Markowitz, H.M., *Mean-Variance Analysis in Portfolio Choice and Capital Markets* , Blackwell Publishing, Oxford, UK, 1987

[75] Marti, K., *Stochastic Optimization Methods*, Springer, 2005

[76] Morgenstern, O., von Neumann, J., *Theory of Games and Economic Behavior*, Princton University Press, 1947

[77] Mathews, G.B., *On the partition of numbers*, Proceedings of the London Mathematical Society 28, 1897, pp. 486-490

[78] Mirkov, R., Pflug, G. CH., *Tree Aproximations of Dynamic Stochastic Programs*, SIAM Journal on Optimization, Volume 18, Issue 3, 2007, pp. 1082-1105

[79] Müller, A., Stoyan, D., *Comparison Methods for Stochastic Models and Risks*, Wiley, New York, 2002

[80] Müschenborn, W., *Interaktive Verfahren zur Lösung linearer Vektoroptimierungsprobleme*, Verlag Harri Deutsch, Frankfurt am Main, 1990

[81] Nemhauser, G.L., Wolsey, L.A., *Integer and Combinatorial Optimization*, Wiley, 1988

[82] Nemirovski, A., Shapiro, A., *On complexity of stochastic programming problems*, Continious Optimization: Current Trends and Applications, (G. Calafiore and F. Dabbene, Eds.), Springer, 2005, pp. 111-144

[83] Nowak, M.P., Römisch, W., *Stochastic Lagrangian relaxation applied to power scheduling in a hydro-thermal system under uncertainty*, Annals of Operations Research 100, 2000, pp. 251-272

[84] Ogryczak, W., Ruszczyński, A., *From Stochastic Dominance to Mean-Risk Models: Semideviations as Risk Measures* , European Journal of Operations Research 116, 1999, pp. 33-50

[85] Ogryczak, W., Ruszczyński, A., *On Consistency of Stochastic Dominance and Mean-Semideviation Models*, Mathematical Programming 89, Series B, 2001, pp. 217-232

[86] Ogryczak, W., Ruszczyński, A., *Dual Stochastic Dominance and Related Mean-Risk Models* , SIAM Journal on Optimization 13, 2002, pp.60-78

[87] Padberg, M., *Linear Optimization and Extensions*, Springer 1995

[88] Papadimitriou, C.H., Steiglitz, K., *Combinatorial Optimization: Algorithms and Complexity*, Prentice-Hall, Englewood Cliffs, New Jersey, 1982

[89] Pflug, G. Ch., *Some remarks on the Value-at-Risk and the Conditional Value-at-Risk* , in Probabilistic Constrained Optimization: Methodology and Applications, (S. Uryasev, Eds.), Kluwer Academic Publishers, Dordrecht, The Netherlands, 2000, pp. 272-281

[90] Pflug, G. Ch., Römisch, W., *Modeling, Measuring and Managing Risk*, World Scientific, Singapore, 2007

[91] Prékopa, A., *Stochastic Programming*, Kluwer, Dordrecht, 1995

[92] Rockafellar, R.T., *Convex Analysis*, Princeton University Press, Princeton, 1997

[93] Rockafellar, R.T., Uryasev, S., *Optimization of Conditional Value-at-Risk* , Journal of Risk 2, 2000, pp. 21-41

[94] Rockafellar, R.T., Uryasev, S., *Conditional Value-at-Risk for General Loss Distributions*, Journal of Banking Finance 26, 2002, pp. 1443-1471

[95] Rockafellar, R.T., Wets, R. J-B., *Variational Analysis*, Springer, Berlin, 1998

[96] Römisch, W., *Stability of Stochastic Programming Problems*, Stochastic Programming, Handbooks in Operations Research and Management Science Vol. 10 (A. Ruszczyński, A. Shapiro, Eds.), Elsevier, Amsterdam, 2003, pp.483-554

[97] Römisch, W., Schultz, R., *Multistage stochastic integer programs: An Introduction* , Online Optimization of Large Scale Systems (M.Grötschel, S.O.Krumke, J.Rambau, Eds.), Springer Berlin, 2001

[98] Ruszczyński, A., *Decomposition methods in stochastic programming* Mathematical Programming 79, 1997, pp. 333-353

[99] Ruszczyński, A., *Decomposition Methods* Stochastic Programming, Handbooks in Operations Research and Management Science Vol. 10 (A. Ruszczyński, A. Shapiro, Eds.), Elsevier, Amsterdam, 2003, pp.141-212

[100] Ruszczyński, A., Shapiro, A. (Eds), *Handbooks in Operations Research and Management Science, 10: Stochastic Programming* , Elsevier, Amsterdam, 2003

[101] Ruszczyński, A., Shapiro, A., *Optimality and Duality in Stochastic Programming*, Stochastic Programming, Handbooks in Operations Research and Management Science Vol. 10 (A. Ruszczyński, A. Shapiro, Eds.), Elsevier, Amsterdam, 2003, pp.65-139

[102] Ruszczyński, A., Shapiro, A., *Conditional Risk Mappings*, Mathematics of Operations Research 31, 2006, pp.544–561

[103] Ruszczyński, A., Shapiro, A., *Optimization of Convex Risk Functions*, Mathematics of Operations Research 31, 2006, pp.433–452

[104] Schrijver, A., *Theory of Linear and Integer Programming*, Wiley, 1986

[105] Schultz, R., *Continuity and stability in two-stage stochastic integer programming*, Stochastic Optimization, Numerical Methods and Technical Applications, Lecture Notes in Economics and Mathematical Systems, Vol. 379 (K. Marti, Eds.), Springer, Berlin, 1992, pp.81-92

[106] Schultz, R., *On structure and stability in stochastic programs with random technology matrix and complete integer recourse*, Mathematical Programming 70, 1995, pp.73-89

[107] Schultz, R., *Some aspects of stability in stochastic programming*, Annals of Operations Research 100, 2000, pp. 55-84

[108] Schultz, R., *Stochastic programming with integer variables* , Mathematical Programming 97, 2003, pp. 285-309

[109] Schultz, R., Tiedemann, S., *Risk aversion via excess probabilities in stochastic programs with mixed-integer recourse* , SIAM Journal of Optimization 14, 2003, pp. 115-138

[110] Schultz, R., Tiedemann, S., *Conditional value-at-risk in stochastic programms with mixed-integer recourse*, Mathematical Programming 105, 2006, pp. 365-386

[111] Sen, S., *Algorithms for Stochastic Mixed-Integer Programming Models*, Disctrete Optimization, Handbooks in Operations Research and Management Science Vol. 12 (Aardal, K., Nemhauser, G., Weismantel, G., Eds.), Elsevier, Amsterdam, 2005, pp.515-558

[112] Shapiro, A., *On complexity of multistage stochastic programs*, Operations Research Letters 34, 2006, pp. 1-8

[113] Shapiro, A., *Quantitative stability in stochastic programming* , Mathematical Programming 67, 1994, pp. 99-108

[114] Tanino, T., Kuk, H., *Nonlinear Multiobjective Programming, In: M. Ehrgott, X. Gandibleux (Eds.), Multiple Criteria Optimization: State of the Art Annoted Bibliographic Surveys*, Kluver, Boston, 2002, pp. 71-128

[115] Tiedemann, S., *Risk Measures with Preselected Tolerance Levels in Two-Stage Stochastic Mixed-Integer Programming*, Dissertation, Institute of Mathematics, University Duisburg-Essen, 2005

[116] Wallace, S.W., Fleten, S.-E., *Stochastic programming models in energy*, Stochastic Programming, Handbooks in Operations Research and Management Science Vol. 10 (A. Ruszczyński, A. Shapiro, Eds.), Elsevier, Amsterdam, 2003, pp.637-688

[117] Wallace, S.W., Ziemba, W.T. (Eds), *Applications of Stochastic Programming*, MPS-SIAM Series in Optimization, 2005

[118] Wang, J., *Stability of multistage stochastic programming*, Annals of Operations Research 56, 1995, pp. 313-322

[119] Weber, S., *Distribution-Invariant Risk Measures, Information, and Dynamic Consistency*, Mathematical Finance 16, 2006, pp. 419-441

[120] Wolsey, L.A., *Integer Programming*, John Wiley Sons Inc, 1998