Adaptive Diskretisierung stochastischer Optimierungsprobleme

Diplomarbeit

Stefan Vigerske

Humboldt-Universität zu Berlin
Mathematisch-Naturwissenschaftliche Fakultät II
Institut für Mathematik

Betreuer: Prof. Dr. W. Römisch

16. Juni 2005

Vorwort

Mehrstufige stochastische Programme sind ein leistungsfähiges Werkzeug um Optimierungsprobleme zu modellieren, bei denen ein Teil der Daten einen stochastischen Prozess bilden. In vielen Fällen ist die zugehörige Wahrscheinlichkeitsverteilung nicht komplett bekannt oder besitzt einen unendlichen Träger. Um solche Optimierungsprobleme numerisch behandeln zu können, muss die Verteilung durch Auswahl repräsentativer Szenarien, d.h. Realisierungen des stochastischen Prozesses, und Zuweisung von Wahrscheinlichkeitsgewichten diskretisiert werden. Solch eine Diskretisierung führt zu einem endlich-dimensionalen Optimierungsproblem, welches aber immer noch zu groß sein kann, um mit aktuellen Lösern behandelt werden zu können. Deshalb muss die Anzahl der Szenarien weiter reduziert werden, so daß das Problem effizient lösbar wird, sich dabei aber nicht zu weit vom Originalproblem entfernt.

Solch ein Szenarioreduktions-Algorithmus ist das Thema dieser Arbeit. Die Reduktion des Szenariobaumes wird dabei in den Lösungsprozess integriert, wobei mit einer groben Approximation des ursprünglichen Szenariobaums begonnen wird und diese so lange verfeinert wird wie nötig. Um die Qualität der berechneten Diskretisierung abzuschätzen, werden Relaxierungstechniken vorgeschlagen.

Der entwickelte Algorithmus wurde für lineare mehrstufige stochastische Programme implementiert. In dieser Arbeit werden numerische Ergebnisse für die Szenarioreduktion aus dem Gebiet des Strommanagements unter unsicherer elektrischer Last präsentiert.

Mein Dank geht an Prof. Werner Römisch und Dr. Ivo Nowak für ihre Unterstützung und ihre Vorschläge, an Holger Heitsch für die Bereitstellung der Strommanagementmodelle und der dazugehörigen Szenariobäume und an viele Freunde für ihre Motivation und Aufmunterung.

Inhaltsverzeichnis

Kapitel 1

Einleitung

Ein stochastisches Programm ist ein Optimierungsproblem, dessen Daten Unsicherheiten unterliegen. Diese Unsicherheit wird mittels einer Zufallsvariable $\xi : \Omega \to \mathbb{R}^s$ über einem Wahrscheinlichkeitsraum $(\Omega, \mathcal{F}, \mathbb{P})$ modelliert. Für eine zweite Zufallsvariable $x : \Omega \to \mathbb{R}^r$ über $(\Omega, \mathcal{F}, \mathbb{P})$, wertet eine Kostenfunktion$f : \mathbb{R}^r \times \mathbb{R}^s \to \mathbb{R}$ die Realisierung $\xi(\omega)$ des Daten-Prozesses und die dazugehörige Realisierung $x(\omega)$ von x aus: $f(\xi(\omega), x(\omega))$. Ziel des Optimierers ist es, einen Entscheidungsprozess x auf Ω zu finden, der $\mathbb{F}[f(x,\xi)]$ minimiert, so daß (s.d.) gewisse Restriktionen an $x(\cdot)$ erfüllt sind. $\mathbb{F}$ repräsentiert dabei oft den Erwartungswert von $f(x,\xi)$ über Ω. Aber auch andere Funktionale über dem linearem Raum der reell-wertigen Zufallsvariablen sind denkbar. In Problemen mit wirtschaftlichen Hintergrund ist $\mathbb{F}$ z.B. oft auch ein Risikomass (vgl. [14, 9]). In dieser Arbeit benutzen wir den Erwartungswert, $\mathbb{F} := \mathbb{E}$, aber Erweiterungen zu allgemeineren Funktionalen sind denkbar. Ein stochastisches Programm lässt sich nun formal schreiben als

$$\begin{array}{lll} \min\limits_{x} & \mathbb{E}[f(x,\xi)] & \\ \text{s.d.} & G(x,\xi) \leq 0 & \mathbb{P}\text{-f.s.} \\ & x \in X(\xi) & \mathbb{P}\text{-f.s.} \end{array} \tag{SP}$$

wobei $G : \mathbb{R}^r \times \mathbb{R}^m \to \mathbb{R}^d$ und $X : \mathbb{R}^m \rightrightarrows \mathbb{R}^r$ die *Restriktionen* für x bestimmen.

Mehrstufige stochastische Programme modellieren eine Folge von sequentiellen stochastischen Optimierungsproblemen: Für einen stochastischen Datenprozess $\xi := \{\xi_t\}_{t=1}^T$ über $(\Omega, \mathcal{F}, \mathbb{P})$ bzgl. der diskreten Zeitstufen $1, \ldots, T$, $\xi_t : \Omega \to \mathbb{R}^{s_t}$, $t = 1, \ldots, T$, $\sum_{t=1}^T s_t = s$, bezeichne $\zeta_t := (\xi_1, \ldots, \xi_t)$ die Zufallsvariable, welche die Ereignisse repräsentiert, die bis zum Zeitpunkt t beobachtbar sind, und $\mathcal{F}_t := \sigma(\zeta_t)$ die Sigma-Algebra, welche von ζ_t, $t = 1, \ldots, T$, erzeugt wird. Wir nehmen an, daß ξ_1 deterministisch ist, d.h. $\mathcal{F}_1 = \{\emptyset, \Omega\}$, und daß zum Zeitpunkt T die gesamte Information bekannt ist: $\mathcal{F}_T = \mathcal{F}$.

Seien

$$\begin{aligned} f : \left(\mathop{\times}_{t=1}^{T} \mathbb{R}^{r_t} \right) \times \left(\mathop{\times}_{t=1}^{T} \mathbb{R}^{s_t} \right) &\to \mathbb{R}, \\ X_t : \mathop{\times}_{\tau=1}^{t} \mathbb{R}^{s_\tau} &\rightrightarrows \mathbb{R}^{r_t}, \qquad t = 1, \ldots, T, \\ g_t : \left(\mathop{\times}_{\tau=1}^{t} \mathbb{R}^{r_\tau} \right) \times \left(\mathop{\times}_{\tau=1}^{t} \mathbb{R}^{s_\tau} \right) &\to \mathbb{R}^{d_t}, \qquad t = 1, \ldots, T, \end{aligned}$$

gegebene Funktionen. Wir suchen nach einem stochastischen Prozess $x := \{x_t\}_{t=1}^T$ über $(\Omega, \mathcal{F}, \mathbb{P})$, $x_t : \Omega \to \mathbb{R}^{r_t}$, $t = 1, \ldots, T$, $\sum_{t=1}^T r_t = r$, von sequentiellen Entscheidungen, welche $\mathbb{E}[f(x,\xi)]$ minimieren, so daß die Restriktionen $g_t(x_1, \ldots, x_t, \zeta_t) \leq 0$, $x_t \in X_t(\zeta_t)$ $\mathbb{P}$-fast sicher für alle $t =$

$1,\ldots,T$ erfüllt sind. Weiterhin fordern wir von $x(\cdot)$, daß die Entscheidung $x_t(\omega)$ zum Zeitpunkt t nur von der Information abhängt, die bis zum Zeitpunkt t beobachtbar ist, d.h. x soll messbar bzgl. $\mathcal{F}_t$ sein. Äquivalente Formulierungen sind: $x_t = \mathbb{E}[x_t | \mathcal{F}_t]$, x_t hängt nur von ζ_t ab, oder x ist nicht-antizipativ. Auf diese Weise hat ein mehrstufiges stochastisches Programm die Form

$$\begin{array}{lllll} \min\limits_{x} & \mathbb{E}[f(x,\xi)] & & & \\ & g_t(x_1,\ldots,x_t,\zeta_t) \leq 0 & \mathbb{P}\text{-f.s.} & t=1,\ldots,T & \\ \text{s.d.} & x_t \in X_t(\zeta_t) & \mathbb{P}\text{-f.s.} & t=1,\ldots,T & \\ & x_t = \mathbb{E}[x_t | \mathcal{F}_t] & & t=1,\ldots,T & \end{array} \qquad \text{(MSP)}$$

Wenn einige der Komponenten von ξ stetig sind, so ist (MSP) unendlich dimensional. Um es numerisch zu lösen, wird es dann durch eine endlich-dimensionale Approximation ersetzt. Auch wenn der Träger von ξ endlich ist, so ist das Problem (MSP) oft nicht computertechnisch behandelbar und muss durch ein Problem kleinerer Dimension approximiert werden. Die Generierung solcher Approximationen wurde in den letzten Jahrzehnten viel untersucht. Oft wird die ursprüngliche Wahrscheinlichkeitsverteilung $\mathbb{P}$ durch eine Approximation $\hat{\mathbb{P}}$ mit endlichem Träger $\Omega_I = \{\omega_i\}_{i\in I}$ ersetzt. Die Realisierungen $\xi_i := \xi(\omega_i)$ von ξ, $i \in I$, repräsentieren ausgewählte Szenarien, und aufgrund der Annahme, daß die Information zum Zeitpunkt 1 deterministisch ist, bilden sie einen Szenariobaum mit Wurzelknoten $\xi_{i,1}$.

Die Herausforderung der Szenariobaum-Erstellung liegt nun in der Auswahl der Szenarien $\{\xi_i\}_{i\in I}$, so daß die Approximation (MSP$[I]$) "in der Nähe" von (MSP) liegt *und* numerisch lösbar ist. Schon die Definition des Abstandes von stochastischen Programmen mit verschiedenen zugrunde liegenden Wahrscheinlichkeitsverteilungen ist eine schwierige Aufgabe.

Dupačová, Römisch, u.a. ([7, 15]) reduzierten die Berechnung des Abstandes zwischen konvexen zwei-stufigen stochastischen Programmen auf ein Maß für den Abstand der darunter liegenden Wahrscheinlichkeitsverteilungen ($\mathbb{P}$ und $\hat{\mathbb{P}}$). Basierend auf Resultaten aus der Stabilitätstheorie (vgl. [20, Kapitel 8]) haben die Autoren gezeigt, daß der Abstand zwischen dem Optimalwert des Originalproblems und dessen Approximation abgeschätzt werden kann durch ein Produkt aus einer problemspezifischen Funktion und dem Abstand der Wahrscheinlichkeitsverteilungen, welcher durch eine Wahrscheinlichkeits-(semi-)metrik (vgl. [18]) gemessen wird. Die vorgeschlagene Methode zur Approximation des Originalproblems ist daher die Auswahl einer Menge von Szenarien vorgegebener Größe, welche so viel Information der Originalverteilung, gemessen am Abstand zu dieser, enthält wie möglich. Da der Abstand von $\mathbb{P}$ und $\hat{\mathbb{P}}$ noch mit einem problemspezifischen Faktor multipliziert wird, ist es schwer, ein quantitatives Mass für die Güte der Approximation von $\mathbb{P}$ so vorzugeben, daß die Störung des Optimalwertes und der optimalen Lösungsmenge klein ist. Der Algorithmus, der in dieser Arbeit entwickelt wird, versucht dieses Problem zu behandeln. Eine detailliertere Diskussion des Szenarioreduktions-Algorithmusses aus [7] findet man daher in Abschnitt 3.1.

Der Algorithmus zur Szenariobaum-Generierung von Casey und Sen für lineare MSPs mit zufälliger rechter Seite ([1]) konstruiert eine Sequenz von Approximationen des MSPs, deren Lösung asymptotisch zu einer Lösung des Originalproblems führt. Für die Lösung x solch eines approximierten Problems wird die Güte der Entscheidung in Form des "Abstandsparameters" δ_n und dem "Unzulässigkeitsindex" η_n für jeden Knoten n des aktuellen Szenariobaums gemessen. Wenn der Abstand oder die Unzulässigkeit eine bestimmte Toleranz für einen Knoten überschreiten, so wird der Baum für diesen Knoten verfeinert. Der Abstandsparameter δ_n ist definiert als der Abstand zwischen einer unteren und einer oberen Schranke einer "salvage value function", welche für jeden Knoten definiert wird und den Optimalwert eines kleinen linearen Programmes darstellt. Die Lösungen dieses Problems können benutzt werden, um eine Entscheidung $\bar{x} = (\bar{x}_n)_n$ für das Originalproblem zu generieren, und es wird gezeigt, daß $\delta_n = 0$ ist für alle Knoten n, vorausgesetzt $\bar{x}$ ist zulässig für das Originalproblem. Der Unzulässigkeitsindex η_n ist eine obere Schranke an die Wahrscheinlichkeit, daß eine Entscheidung x_n unzulässig ist. Der Vorteil dieses Algorithmusses ist seine asymptotische Konvergenz, sein Optimalitätsmass, und die Möglichkeit, die Wahrscheinlichkeit zu kontrollieren, daß eine für den reduzierten Szenariobaum generierte Lösung $\bar{x}$ zulässig ist für jeden Knoten des Originalbaumes. Die computertechnische Herausforderung besteht in der Lösung des approximierten Problemes in jeder Iteration und der Bewertung der Güte der erhaltenen Entscheidungen in jedem Knoten des Szenariobaumes. Letzteres scheint hervorragend für Parallelcomputer geeignet zu sein.

Ein anderer Algorithmus zur Szenariobaum-Generierung, welcher nodale Information benutzt, ist der EVPI-basierte "importance sampling" Algorithmus von Dempster und Thompson ([4]). Der Erwartungswert der perfekten Information (EVPI) eines stochastischen Programmes wird zunächst berechnet durch das Finden einer optimalen Lösung für jede mögliche Realisierung der Zufallsvariablen und der Bildung des Erwartungswertes dieser Lösungen, d.h. das Problem (MSP) wird gelöst, wobei die Nichtantizipativitätsbedingungen ignoriert werden. Danach wird der Optimalwert des wahren Problemes subtrahiert. Der EVPI, normalisiert durch den Optimalwert des Problemes, wird benutzt als ein Mass für die Unsicherheit im Modell. Das Konzept des EVPIs eines stochastischen Programmes kann erweitert werden auf den sogenannten lokalen EVPI eines Knotens des Szenariobaumes: Für einen Knoten n des Szenariobaumes ist der lokale EVPI als die Differenz zwischen dem Optimalwert des Problemes mit Wurzelknoten n und dem Erwartungswert im Fall, daß perfekte Information über die Zukunft verfügbar ist, d.h. der gewichtete Mittelwert der Optimalwerte entlang jedes Szenarios, welches Knoten n passiert, definiert. Der EVPI-basierte Sampling-Algorithmus generiert iterativ Szenariobäume durch Erhöhung der Anzahl von Verzweigungen an Knoten mit relativ hohem EVPI, durch Resampling von Verzweigungen an Knoten mit vernachlässigbarem EVPI und durch Verringerung der Anzahl von Verzweigungen an Knoten, welche wiederholt einen relativ kleinen EVPI haben. Dieser Algorithmus wurde für lineare mehrstufige stochastische Programme im Softwarepaket MSLiP implementiert, welches besonders die Parallelisierbarkeit der benutzten Algorithmen ausnutzt. Das lineare MSP welches zu einem generierten Szenariobaum gehört wird mit einem "nested Benders decomposition" Algorithmus ([5]), einem Schnittebenenverfahren basierend auf der Relaxierung von zeitkoppelnden Restriktionen, gelöst. Eine kurze Einführung in diese Dekompositionsmethode ist in Abschnitt 3.6.4 zu finden.

Andere Ansätze zur Lösung von MSPs basieren auf dualen Dekompositionstechniken. Edirisinghe's Algorithmus ([8]) für lineare MSPs relaxiert die Nichtantizipativitätsforderungen an den Entscheidungsprozess bis auf eine bestimmte Aggregierung zweiter Ordnung und die Berechnung von unteren Schranken. Um die Aggregierung zu bestimmen, wird angenommen, daß der Träger Ω von ξ in ein kompaktes Simplex $\hat{\Omega}$ eingeschlossen werden kann, welches dann in kleinere Teilsimplizes partitioniert wird. In jeder Iteration wird die aktuelle Aggregierung genutzt, um untere und obere Schranken für den Optimalwert zu bestimmen. Dann wird ein Teilsimplex mit großem Beitrag an dem Abstand zwischen oberer und unterer Schranke ausgewählt und verfeinert. Der Vorteil dieses Ansatzes ist die Berechnung von Schranken an den Optimalwert des Originalproblemes und die Möglichkeit, parallele Computerumgebungen auszunutzen. Im Moment ist es aber unklar, ob diese Methode Garantien bzgl. einer asymptotischen Konvergenz liefern kann.

Higle, Rayco und Sen präsentierten in [16] einen stochastischen Szenariodekompositionsalgorithmus, welcher nicht von einem endlichen Träger von ξ oder Linearität des MSPs abhängig ist. Die Verteilung $\mathbb{P}$ wird durch eine empirische Verteilung approximiert, die man mittels Szenario-Sampling erhält. Das duale Problem, entstanden durch Dualisierung der Nichtantizipativitätsbedingungen für die empirische Verteilung, wird approximiert mittels eines Schnittebenenverfahrens, welches das sogenannte Master-Problem generiert. In jeder Iteration liefern Lösungen der Teilprobleme einen neuen Schnitt, um das Master-Problem zu aktualisieren. Die dualen Multiplikatoren werden dann durch Lösung des Master-Problemes neu bestimmt, und die empirische Verteilung wird durch Sampling eines neuen Szenarios aktualisiert. Um ein zu starkes Wachstum des Master-Problemes durch die Anzahl der generierten Szenarien und Schnitte zu verhindern, wird ein Variablenaggregierungsschema vorgeschlagen, welches solche Spalten aggregiert, deren zugrunde liegende Daten im Problem ähnlich sind. Wir werden eine formalere Beschreibung des Algorithmusses am Ende von Abschnitt 3.4.1 geben. Vorteile dieses Ansatzes sind seine milden Voraussetzungen an das stochastische Programm und die Möglichkeit, mit einer hohen Anzahl von Szenarien zu arbeiten. Auf der anderen Seite sind keine Abschätzungen für die Qualität der Lösung nach einer endlichen Zahl von Iterationen bekannt, d.h. es ist nicht möglich mit einer Optimalitätstoleranz zu arbeiten. Fast sichere asymptotische Konvergenz kann aber gezeigt werden.

In dieser Arbeit wird ein Algorithmus entwickelt, welcher die Konstruktion des reduzierten Szenariobaumes in den Lösungsprozess des MSPs integriert, und dabei nur milde Anforderungen an die Funktionen f, g_t und X_t stellt. Der Ursprung dieses Algorithmusses ist eine Idee von Nowak, vgl. [17, Kapitel 9]. Beginnend mit einer groben Aggregierung eines gegebenen Szenariobaumes wird diese Aggregierung so lange verfeinert wie der geschätzte Abstand zum Originalproblem zu groß ist. Die Aggregierungen werden mittels dem Fast-Forward-Selection Algorithmus aus [15] gebildet. Schranken an den Optimalwert des approximierten MSPs werden durch Lösen einer konvexen Relaxierung berechnet. Weiterhin liefert die Lösung einer konvexen Relaxierung eine Approximierung für einen Optimalpunkt des Originalproblemes. Wenn der Algorithmus die Generierung des Szenariobaumes abgeschlossen hat, kann eine lokale Suche für das diskretisierte MSP gestartet werden, ausgehend von der letzten Lösung der konvexen Relaxierung. Zur Aktualisierung der Relaxierung wird ein Spaltengenerierungsalgorithmus, basierend auf Lagrange-Dekomposition, skizziert. Numerische Ergebnisse für die Optimierung der Betriebsniveaus einer Menge von kohlebetriebenen Wärmekraftwerken und der Füllstände von Pumpwasserspeicherwerken unter unsicherer Last werden präsentiert.

In Kapitel 2 formulieren wir den Algorithmus in Form eines adaptiven Diskretisierungsalgorithmus für unendlich-dimensionale MINLPs. Viel Notation, um das Originalproblem, seine Diskretisierung und Aggregierung, und die Dekompositions-, Relaxierungs- und Verfeinerungstechniken zu beschreiben, wird dort eingeführt. Kapitel 3 beginnt mit einem Überblick von [7] und fährt dann damit fort, den abstrakten Algorithmus aus Kapitel 2 auf mehrstufige stochastische Programme zu übertragen. Die Szenariobaumform eines MSPs wird formuliert, und Algorithmen für die Aggregierung und die Verfeinerung eines Szenariobaumes werden eingeführt. Weiterhin werden Dekompositionstechniken, welche die spezielle Struktur mehrstufiger stochastischer Programme ausnutzen, vorgestellt. Als einen Spezialfall behandeln wir gemischt-ganzzahlige lineare MSPs. Das vierte Kapitel beschreibt eine Implementierung des vorgestellten Algorithmusses für lineare MSPs. Numerische Ergebnisse für das Strommanagement in einem hydro-thermalen System unter Unsicherheit werden präsentiert. Wir beenden die Arbeit mit Verbesserungsvorschlägen und einigen abschließenden Bemerkungen.

Kapitel 2

Ein abstrakter adaptiver Diskretisierungsalgorithmus

Dieses Kapitel gibt eine abstrakte Formulierung eines adaptiven Diskretisierungsalgorithmusses für unendlich-dimensionale gemischt-ganzzahlige nichtlineare Optimierungsprobleme. Zunächst führen wir die notwendigen Notationen ein, danach skizzieren wir den optimalen Verfeinerungsalgorithmus. Weiterhin konzentrieren wir uns auf Algorithmen zur Aktualisierung von konvexen Relaxierungen, welche vom Verfeinerungsalgorithmus benutzt werden. Letztendlich formulieren wir den adaptiven Diskretisierungalgorithmus.

2.1 Das Problem

Wir betrachten Optimierungsprobleme der Form

$$\begin{array}{ll} \min\limits_{x \in E} & F(x) \\ \text{s.d.} & \begin{array}{r} G(x) \leq 0 \\ x \in X \end{array} \end{array} \tag{P}$$

wobei E einen Banachraum beschreibt, möglicherweise unendlich-dimensional, $G : E \to \mathbb{R}$ und $X \subseteq E$.

Als ein Beispiel betrachte das stochastische Programm (SP) und sei E ein Banachraum von meßbaren Funktionen, $F(x) := \mathbb{E}[f(x,\xi)]$, $X := \{x : \Omega \to \mathbb{R}^r \,|\mathbb{P}[x \in X(\xi)] = 1\}$, und $G(x) := \mathbb{P}[G(x,\xi) > 0]$.

Eine konvexe Relaxierung (C) von (P) hat die Form

$$\begin{array}{ll} \min\limits_{x} & \check{F}(x) \\ \text{s.d.} & \begin{array}{rl} \check{G}(x) \leq 0 & \\ x \in \check{X} & \\ c(x) \leq 0 & c \in \mathcal{C} \end{array} \end{array} \tag{C}$$

wobei $\check{X}$ eine konvexe Obermenge von X ist, $\check{F}$ und $\check{G}$ sind konvexe Unterschätzerfunktionen von F und G über $\check{X}$ und $\mathcal{C} \subset \{c : E \to \mathbb{R} \text{ konvex}\}$ ist eine Menge von Schnitten, welche hinzugefügt werden um die Lücke $|\mathrm{val}(P) - \mathrm{val}(C)|$ zwischen den Optimalwerten von (P) und (C) (falls diese existieren) zu reduzieren.

Eine Diskretisierung von (P) basiert auf der endlichen Knotenmenge $\bar{N}$:

$$\begin{array}{ll} \min\limits_{x} & F_{\bar{N}}(x) \\ \text{s.d.} & G_{\bar{N}}(x) \leq 0 \\ & x \in X_{\bar{N}} \end{array} \qquad (P[\bar{N}])$$

wobei $x = (x_n)_{n\in\bar{N}} \in \mathbb{R}^{r_{\bar{N}}}$, $F_{\bar{N}} : \mathbb{R}^{r_{\bar{N}}} \to \mathbb{R}$, $G_{\bar{N}} : \mathbb{R}^{r_{\bar{N}}} \to \mathbb{R}^{d_{\bar{N}}}$, $X_{\bar{N}} \subseteq \mathbb{R}^{r_{\bar{N}}}$.

Die konvexe Relaxierung von $(P[\bar{N}])$ ist

$$\begin{array}{lll} \min\limits_{x} & \check{F}_{\bar{N}}(x) & \\ \text{s.d.} & \check{G}_{\bar{N}}(x) \leq 0 & \\ & x \in \check{X}_{\bar{N}} & \\ & c(x) \leq 0 & c \in \mathcal{C}_{\bar{N}} \end{array} \qquad (C[\bar{N}])$$

wobei $\check{X}_{\bar{N}}$ eine konvexe Obermenge von $X_{\bar{N}}$ ist, $\check{F}_{\bar{N}}$ und $\check{G}_{\bar{N}}$ sind konvexe Unterschätzerfunktionen von $F_{\bar{N}}$ und $G_{\bar{N}}$ über $\check{X}_{\bar{N}}$ und $\mathcal{C}_{\bar{N}} \subset \{c : \mathbb{R}^{r_{\bar{N}}} \to \mathbb{R} \text{ konvex}\}$ ist eine Menge von Schnitten.

2.2 Aggregierung

Wir betrachten nun *Partitionen* $\mathcal{N}$ der Knotenmenge $\bar{N}$ (Aggregierungen von Knoten), d.h. $\mathcal{N}$ ist eine Familie von Mengen $N \subseteq \bar{N}$, so daß $\bigcup\mathcal{N} := \bigcup_{N\in\mathcal{N}} N = \bar{N}$ und $N \cap M = \emptyset$ für alle $N, M \in \mathcal{N}$, $N \neq M$. Das aggregierte Problem ist definiert für einen Vektor $x = (x_N)_{N\in\mathcal{N}}$. Diese Aggregierung könnte durch Mittelung über die Knoten einer Menge N ($x_N := \sum_{n\in N} w_n x_n$, $w_n \geq 0$, $n \in N$, $\sum_{n\in N} w_n = 1$) oder durch Auswahl eines repräsentativen Knotens aus N ($x_N := x_n$ für ein ausgezeichnetes $n \in N$) konstruiert werden. Führt man die Aggregierung in die Funktionen $F_{\bar{N}}$, $G_{\bar{N}}$ und die Menge $X_{\bar{N}}$ ein, so erhält man Funktionen $F_{\mathcal{N}} : \mathbb{R}^{r_{\mathcal{N}}} \to \mathbb{R}$ und $G_{\mathcal{N}} : \mathbb{R}^{r_{\mathcal{N}}} \to \mathbb{R}^{d_{\mathcal{N}}}$ und die Menge $X_{\mathcal{N}} \subset \mathbb{R}^{r_{\mathcal{N}}}$. Auf ähnliche Weise nehmen wir an, daß eine konvexe Obermenge $\check{X}_{\mathcal{N}}$ von $X_{\mathcal{N}}$ und konvexe Unterschätzerfunktionen $\check{F}_{\mathcal{N}}$ und $\check{G}_{\mathcal{N}}$ von $F_{\mathcal{N}}$ und $G_{\mathcal{N}}$ über $\check{X}_{\mathcal{N}}$ existieren.

Dann liest sich das aggregierte Problem

$$\begin{array}{ll} \min\limits_{x} & F_{\mathcal{N}}(x) \\ \text{s.d.} & G_{\mathcal{N}}(x) \leq 0 \\ & x \in X_{\mathcal{N}} \end{array} \qquad (P[\mathcal{N}])$$

Die konvexe Relaxierung von $(P[\mathcal{N}])$ ist

$$\begin{array}{lll} \min\limits_{x} & \check{F}_{\mathcal{N}}(x) & \\ \text{s.d.} & \check{G}_{\mathcal{N}}(x) \leq 0 & \\ & x \in \check{X}_{\mathcal{N}} & \\ & c(x) \leq 0 & c \in \mathcal{C}_{\mathcal{N}} \end{array} \qquad (C[\mathcal{N}])$$

wobei $\mathcal{C}_{\mathcal{N}}$ eine Menge zulässiger Schnitte bezeichne.

2.3 Optimale Verfeinerung

Notation. Eine *Verfeinerung* einer Partitionierung $\mathcal{N}$ von $\bar{N}$ ist eine Partitionierung $\mathcal{N}'$ von $\bar{N}$, so daß für alle $N' \in \mathcal{N}'$ ein $N \in \mathcal{N}$ mit $N' \subseteq N$ existiert. Wir schreiben dann $\mathcal{N}' \preceq \mathcal{N}$. Wenn zusätzlich $|\mathcal{N}'| > |\mathcal{N}|$, so schreiben wir $\mathcal{N}' \prec \mathcal{N}$.

Für eine Verfeinerung $\mathcal{N}'$ von $\mathcal{N}$ und eine Menge $N \in \mathcal{N}$ sei $\mathcal{N}'|_N := \{N' \in \mathcal{N}' \,|\, N' \subseteq N\}$ die Beschränkung von $\mathcal{N}'$ auf N. Beachte, daß $\bigcup\mathcal{N}'|_N = N$.

Gegeben sei eine große Knotenmenge $\bar{N}$ und eine initiale (triviale) Partitionierung $\mathcal{N}_0 := \{\bar{N}\}$ von $\bar{N}$. Wir möchten eine Diskretisierung $(P[\mathcal{N}])$ mit $|\mathcal{N}| \ll |\bar{N}|$ und kleinem Approximationsfehler

$$\left|\operatorname{val}\left(P\left[\bar{N}\right]\right) - \operatorname{val}\left(P\left[\mathcal{N}\right]\right)\right|$$

konstruieren. Sei dazu $\mathcal{N}' \prec \mathcal{N}$ eine Verfeinerung von $\mathcal{N}$. Dann zerlegen wir den Approximationsfehler in die Fehler, die durch $\mathcal{N}$ als eine Approximation von $\mathcal{N}'$ und durch $\mathcal{N}'$ (bzw. $\bar{N}|_{\mathcal{N}'}$) als eine Approximation von $\bar{N}$ eingeführt werden. Für den ersten Fehler betrachten wir den Abstand der Optimalwerte der konvexen Relaxierungen bzgl. $\mathcal{N}$ und $\mathcal{N}'$, für den zweiten Fehler nehmen wir zunächst nur an, daß ein Mass für die Feinheit der Aggregierung $\mathcal{N}'$ vorliegt. Da wir mit konvexen Relaxierungen arbeiten, ergeben sich weiterhin Fehlerterme bzgl. dem Abstand der Optimalwerte eines Problems und seiner Relaxierung. Insgesamt, schätzen wir den Fehler $\left|\operatorname{val}\left(P\left[\bar{N}\right]\right) - \operatorname{val}\left(P\left[\mathcal{N}\right]\right)\right|$ also durch

$$\begin{aligned}\left|\operatorname{val}\left(P\left[\bar{N}\right]\right) - \operatorname{val}\left(P\left[\mathcal{N}'\right]\right)\right| &+ \left|\operatorname{val}\left(P\left[\mathcal{N}'\right]\right) - \operatorname{val}\left(C\left[\mathcal{N}'\right]\right)\right| \\ &+ \left|\operatorname{val}\left(C\left[\mathcal{N}'\right]\right) - \operatorname{val}\left(C\left[\mathcal{N}\right]\right)\right| + \left|\operatorname{val}\left(C\left[\mathcal{N}\right]\right) - \operatorname{val}\left(P\left[\mathcal{N}\right]\right)\right|\end{aligned}$$

ab. Um die Lücke $\left|\operatorname{val}\left(C\left[\mathcal{N}\right]\right) - \operatorname{val}\left(C\left[\mathcal{N}'\right]\right)\right|$ zu verkleinern, untersuchen wir die "Wichtigkeit" einer Verfeinerung $\mathcal{N}'|_N$ von $N \in \mathcal{N}$ hinsichtlich dieser Lücke. Die Verfeinerung solcher $N \in \mathcal{N}$, welche eine hohe Verbesserung versprechen, werden behalten, während die anderen verworfen werden. Um die Lücken $\left|\operatorname{val}\left(P\left[\mathcal{N}'\right]\right) - \operatorname{val}\left(C\left[\mathcal{N}'\right]\right)\right|$ und $\left|\operatorname{val}\left(P\left[\mathcal{N}\right]\right) - \operatorname{val}\left(C\left[\mathcal{N}\right]\right)\right|$ zu verkleinern, schlagen wir im nächsten Abschnitt einen Spaltengenerierungsalgorithmus zur Verbesserung der Relaxierung vor, welcher eine Reduktion bis auf die Dualitätslücke erlaubt. Die Lücke $\left|\operatorname{val}\left(P\left[\bar{N}\right]\right) - \operatorname{val}\left(P\left[\mathcal{N}'\right]\right)\right|$ wird kontrolliert durch Untersuchung des Abstandes zwischen der Aggregierung $\mathcal{N}'$ und $\bar{N}$. Da dies von der aktuellen Anwendung abhängt, nehmen wir hier nur an, daß ein Maß $\mathcal{D}(\mathcal{N})$ für den Abstand zwischen $\bar{N}|_{\mathcal{N}}$ und $\bar{N}$ zur Verfügung steht.

Wir konzentrieren uns im folgenden auf die Lücke $\left|\operatorname{val}\left(C\left[\mathcal{N}\right]\right) - \operatorname{val}\left(C\left[\mathcal{N}'\right]\right)\right|$: Sei $\mathcal{N}$ eine Partitionierung von $\bar{N}$, $\mathcal{N}' \prec \mathcal{N}$ eine Verfeinerung von $\mathcal{N}$, und sei $\mathcal{M} \subseteq \mathcal{N}$ eine Menge von Teilmengen von $\bar{N}$ deren Verfeinerung wir behalten wollen. Bezeichne dann mit

$$\mathcal{N}(\mathcal{M}) := \{\mathcal{N}'|_M \mid M \in \mathcal{M}\} \cup (\mathcal{N} \setminus \mathcal{M}) \tag{$\mathcal{N}(\mathcal{M})$}$$

die Verfeinerung von $\mathcal{N}$ in Bezug auf $\mathcal{M}$, d.h. die Mengen $M \in \mathcal{M}$ werden ersetzt durch ihre Verfeinerung $\mathcal{N}'|_M$, während die verbleibenden Mengen $\mathcal{N} \setminus \mathcal{M}$ unberührt bleiben.

In jedem Schritt der Methode wird eine Menge $\mathcal{M} \subseteq \mathcal{N}$ mit $|\mathcal{M}| \leq \bar{m}$ bestimmt, so daß

$$\left|\operatorname{val}\left(C\left[\mathcal{N}(\mathcal{M})\right]\right) - \operatorname{val}\left(C\left[\mathcal{N}'\right]\right)\right|$$

klein ist. Wir berechnen die Menge $\mathcal{M}$ mittels folgender Strategie: Sei σ_P eine gegebene Toleranz für $\left|\operatorname{val}\left(C\left[\mathcal{N}'\right]\right) - \operatorname{val}\left(C\left[\mathcal{N}\right]\right)\right|$, relativ zu $\operatorname{val}\left(C\left[\mathcal{N}'\right]\right)$. Nachdem die Probleme $(C[\mathcal{N}])$ und $(C[\mathcal{N}'])$ gelöst wurden, kann entweder $\operatorname{val}(C[\mathcal{N}])$ viel kleiner als $\operatorname{val}(C[\mathcal{N}'])$ sein, $\operatorname{val}(C[\mathcal{N}'])$ kann viel kleiner als $\operatorname{val}(C[\mathcal{N}])$ sein oder beide Werte sind ausreichend nah zueinander (relativ zur Toleranz σ_P). Wir gehen nun folgendermaßen vor:

1. Ist $\operatorname{val}(C[\mathcal{N}']) - \operatorname{val}(C[\mathcal{N}]) > \sigma_P \cdot |\operatorname{val}(C[\mathcal{N}])|$, also $\operatorname{val}(C[\mathcal{N}])$ viel kleiner als $\operatorname{val}(C[\mathcal{N}'])$, so wählen wir $\mathcal{M}$ als die $N \in \mathcal{N}$ welche den höchsten Anteil an der Differenz $\operatorname{val}(C[\mathcal{N}']) - \operatorname{val}(C[\mathcal{N}])$ haben. Da wir keinerlei Voraussetzungen an die Struktur von $F_{\mathcal{N}}$ bzw. $\check{F}_{\mathcal{N}}$ gemacht haben, ist es schwierig eine Formel für die Berechnung von $\mathcal{M}$ in diesem Fall anzugeben. Als ein Beispiel, nehmen wir an, $\check{F}_{\mathcal{N}}$ kann zerlegt werden in eine Summe $\sum_{N \in \mathcal{N}} \check{F}_N$, d.h. für $x = (x_N)_{N \in \mathcal{N}}$ sei
$$\check{F}_{\mathcal{N}}(x) = \sum_{N \in \mathcal{N}} \check{F}_N(x_N).$$

Für $x \in \operatorname{Argmin}(C[\mathcal{N}])$ und $x' \in \operatorname{Argmin}(C[\mathcal{N}'])$ sei dann

$$\mathcal{M} := \operatorname{argmax}\left\{ \sum_{M \in \mathcal{M}} \left(\sum_{M' \in \mathcal{N}'|_M} \check{F}_{M'}(x') \right) - \check{F}_M(x) \,\middle|\, \mathcal{M} \subseteq \mathcal{N},\ |\mathcal{M}| \le \bar{m} \right\}. \tag{2.1}$$

Da $\operatorname{val}(C[\mathcal{N}]) - \operatorname{val}(C[\mathcal{N}']) = \sum_{N \in \mathcal{N}} \left(\check{F}_N(x) - \sum_{N' \in \mathcal{N}'|_N} \check{F}_{N'}(x') \right) < 0$ ist, gilt $\mathcal{M} \neq \emptyset$.

2. Ist $\operatorname{val}(C[\mathcal{N}]) - \operatorname{val}(C[\mathcal{N}']) > \sigma_P \cdot |\operatorname{val}(C[\mathcal{N}'])|$, also $\operatorname{val}(C[\mathcal{N}])$ viel größer als $\operatorname{val}(C[\mathcal{N}'])$, so müssen wir Mengen $N \in \mathcal{N}$ finden, deren Verfeinerung eine große Verkleinerung von $\operatorname{val}(C[\mathcal{N}])$ erlauben. Aber anstatt ähnlich wie im Fall 1 zu verfahren, welcher die Restriktionen $G_{\mathcal{N}}$ und $X_{\mathcal{N}}$ nicht direkt einbezogen hat, wählen wir hier eine Methode, die weniger "greedy" ist.

   Eigentlich wollen wir $\operatorname{val}(C[\mathcal{N}]) - \operatorname{val}(C[\mathcal{N}']) \le \sigma_P |\operatorname{val}(C[\mathcal{N}'])|$ erhalten. Dies kann erreicht werden, indem wir einen Punkt $x = (x_N)_{N \in \mathcal{N}}$ suchen, der zulässig für $(C[\mathcal{N}'])$ ist und sich in der Nähe der optimalen Lösungsmengen von $(C[\mathcal{N}'])$ befindet. Um die Mengen $N \in \mathcal{N}$ zu finden, die dieses Ziel behindern, berechnen wir einen Punkt $x' = (x'_{N'})_{N' \in \mathcal{N}'}$, welcher zulässig für $(C[\mathcal{N}'])$ ist, dessen Zielfunktionswert nicht den Wert von $\operatorname{val}(C[\mathcal{N}'])$ um mehr als den Anteil σ_P überschreitet, und der sich so "nahe" an der Aggregierung $\mathcal{N}$ befindet, wie möglich. Dann wählen wir die $N \in \mathcal{N}$, für die der Anteil am Abstand von x' zur Aggregierung $\mathcal{N}$ am größten ist.

   Etwas formaler, um eine künstliche Restriktion von $\mathcal{N}'$ auf $\mathcal{N}$ auszudrücken, nehmen wir an, daß Matrizen W_N, $N \in \mathcal{N}$, existierten, so daß ein Punkt $x' = (x'_{N'})_{N' \in \mathcal{N}'}$, welcher $W_N x'_N = 0$, $N \in \mathcal{N}$ ($x'_N := (x'_{N'})_{N' \in \mathcal{N}'|_N}$), erfüllt, eine eindeutige Darstellung als ein Punkt $x = (x_N)_{N \in \mathcal{N}}$ besitzt. Die Matrizen W_N drücken Mittelungs- oder Gleichheitsbedingungen an x'_N aus, im Einklang mit der Definition von $F_{\mathcal{N}}$, $G_{\mathcal{N}}$ und $X_{\mathcal{N}}$. Zum Beispiel können Gleichheitsbedingungen mittels $W_N x'_N := (x'_{M'} - x'_{N'})_{N' \in \mathcal{N}'|_N}$ für eine ausgezeichnete Menge $M' \in \mathcal{N}'|_N$, $N \in \mathcal{N}$, ausgedrückt werden.

   Wir suchen nun einen Punkt x', welcher zulässig für $(C[\mathcal{N}'])$ ist, der $\check{F}_{\mathcal{N}'}(x') \le \operatorname{val}(C[\mathcal{N}']) + \sigma_P |\operatorname{val}(C[\mathcal{N}'])|$ erfüllt und die Verletzung der $W_N x'_N = 0$ minimiert, $N \in \mathcal{N}$. Wir lösen also das konvexe Problem

$$\begin{array}{lll} \min\limits_{x'} & \sum\limits_{N \in \mathcal{N}} \|W_N x'_N\|_\infty & \\ \text{s.d.} & \check{G}_{\mathcal{N}'}(x') \le 0 & \\ & x' \in \check{X}_{\mathcal{N}'} & \\ & c(x') \le 0 & c \in \mathcal{C}_{\mathcal{N}'} \\ & \check{F}_{\mathcal{N}'}(x') \le \operatorname{val}(C[\mathcal{N}']) + \sigma_P |\operatorname{val}(C[\mathcal{N}'])| & \end{array} \tag{Proj$[\mathcal{N},\mathcal{N}']$}$$

   Im Falle $\operatorname{val}(\operatorname{Proj}[\mathcal{N},\mathcal{N}']) = 0$, sind wir fertig, da wir einen Punkt x' mit $\check{F}_{\mathcal{N}'}(x') \le \operatorname{val}(C[\mathcal{N}']) + \sigma_P |\operatorname{val}(C[\mathcal{N}'])|$ und $W_N x'_N = 0$ für alle $N \in \mathcal{N}$ gefunden haben. Nehmen wir an, daß der zugehörige Punkt $x = (x_N)_{N \in \mathcal{N}}$ zulässig ist für $(C[\mathcal{N}])$ (weil $x' \in \operatorname{feas}(C[\mathcal{N}'])$ und $W_N x_N = 0$, $N \in \mathcal{N}$) und daß $\check{F}_{\mathcal{N}'}(x') = \check{F}_{\mathcal{N}}(x)$, so kann dieser Fall nicht auftreten, denn er impliziert $\operatorname{val}(C[\mathcal{N}]) - \operatorname{val}(C[\mathcal{N}']) \le \sigma_P |\operatorname{val}(C[\mathcal{N}'])|$, im Widerspruch zur Annahme.

   Ist anderseits $\operatorname{val}(\operatorname{Proj}[\mathcal{N},\mathcal{N}']) > 0$, so fügen wir die $N \in \mathcal{N}$ mit der größten Verletzung der $W_N x'_N = 0$ Restriktionen zu $\mathcal{M}$ hinzu, d.h. für $x^* \in \operatorname{Argmin}(\operatorname{Proj}[\mathcal{N},\mathcal{N}'])$ sei

$$\mathcal{M} := \operatorname{argmax}\left\{ \sum_{M \in \mathcal{M}} \|W_M x^*_M\|_\infty \,\middle|\, \mathcal{M} \subseteq \mathcal{N},\ |\mathcal{M}| \le \bar{m},\ \|W_M x^*_M\|_\infty > 0\, \forall M \in \mathcal{M} \right\}. \tag{2.2}$$

3. Andernfalls gilt

$$\frac{\mathrm{val}\left(C\left[\mathcal{N}'\right]\right) - \mathrm{val}\left(C\left[\mathcal{N}\right]\right)}{\left|\mathrm{val}\left(C\left[\mathcal{N}\right]\right)\right|} \leq \sigma_P \quad \text{und} \quad \frac{\mathrm{val}\left(C\left[\mathcal{N}\right]\right) - \mathrm{val}\left(C\left[\mathcal{N}'\right]\right)}{\left|\mathrm{val}\left(C\left[\mathcal{N}'\right]\right)\right|} \leq \sigma_P.$$

   Infolgedessen ist die Lücke $\left|\mathrm{val}\left(C\left[\mathcal{N}\right]\right) - \mathrm{val}\left(C\left[\mathcal{N}'\right]\right)\right|$ klein genug und wir können mit der Kontrolle von $\mathcal{D}\left(\mathcal{N}'\right)$, dem Abstand der Aggregierung $\mathcal{N}'$ zu $\bar{N}$, fortfahren.

Nachdem Teile der Verfeinerung $\mathcal{N}'$ nach $\mathcal{N}$ entsprechend Formel ($\mathcal{N}\left(\mathcal{M}\right)$) übernommen wurden, brauchen wir eine neue Test-Verfeinerung $\mathcal{N}'$ für die nächste Iteration. Wir berechnen diese durch Partitionierung der $M' \in \mathcal{N}'$ für die ein $M \in \mathcal{M}$ mit $M' \subseteq M$ existiert. Wir definieren also

$$\mathcal{N}'\left(\mathcal{M}\right) := \left(\bigcup_{N \in \mathcal{N} \setminus \mathcal{M}} \mathcal{N}'|_N\right) \cup \left(\bigcup_{M \in \mathcal{M}} \bigcup_{M' \in \mathcal{N}'|_M} \mathrm{part}\left(M'\right)\right). \tag{$\mathcal{N}'\left(\mathcal{M}\right)$}$$

Die Berechnung von $\mathrm{part}\left(M'\right)$ und $W_{M''}$, $M'' \in \mathrm{part}\left(M'\right)$, ist problemabhängig. Möglich wäre es für eine Metrik $d : \bar{N} \times \bar{N} \to \mathbb{R}_+$ eine Partitionierung von $M' \subseteq \bar{N}$ mittels eines Clustering-Algorithmusses zu berechnen.

Die Größe $\sum_{M' \in \mathcal{N}'|_M} \left|\mathrm{part}\left(M'\right)\right|$ für $M \in \mathcal{M}$ lässt sich mittels $\left\|W_M x_M\right\|_\infty$ kontrollieren, d.h. der Anteil von $\sum_{M' \in \mathcal{N}'|_M} \left|\mathrm{part}\left(M'\right)\right|$ an $\sum_{M \in \mathcal{M}} \sum_{M' \in \mathcal{N}'|_M} \left|\mathrm{part}\left(M'\right)\right|$ könnte (ungefähr)

$$\frac{\left\|W_M x_M\right\|_\infty}{\sum_{M \in \mathcal{M}} \left\|W_M x_M\right\|_\infty} \tag{2.3}$$

sein.

Der nächste Schritt ist es, aufzuzeigen, wie die Lücke $\left|\mathrm{val}\left(P\left[\mathcal{N}\right]\right) - \mathrm{val}\left(C\left[\mathcal{N}\right]\right)\right|$ kontrolliert werden kann. Dies geschieht im nächsten Abschnitt. Danach können wir den adaptiven Verfeinerungsalgorithmus in seiner vollen Form darstellen. Für eine Zusammenfassung der oberen Überlegungen, welche den Kern des letztendlichen Algorithmusses bilden, geben wir hier einen ersten Entwurf des Algorithmusses an, welcher aber noch die Aktualisierung der konvexen Relaxierung vermissen lässt:

Algorithmus 2.1 (adaptive Diskretisierung (Entwurf)).
Input: initiale Knotenmenge $\bar{N}$, Parameter $\bar{m} \in \mathbb{N}$ und $\sigma_P \in [0,1]$, $\sigma_D \in [0,1]$
Compute:
  bestimme anfängliche grobe Aggregierungen $\mathcal{N}$ und $\mathcal{N}'$ von $\bar{N}$, $\mathcal{N}' \preceq \mathcal{N}$
  $\mathcal{D}_0 := \mathcal{D}\left(\mathcal{N}'\right)$
  repeat
    löse $(C\left[\mathcal{N}\right])$ und $(C\left[\mathcal{N}'\right])$
    if $\mathrm{val}\left(C\left[\mathcal{N}'\right]\right) - \mathrm{val}\left(C\left[\mathcal{N}\right]\right) > \sigma_P \cdot \left|\mathrm{val}\left(C\left[\mathcal{N}\right]\right)\right|$ **then**
      bestimme $\mathcal{M}$ gemäß Formel (2.1)
      bestimme $\mathcal{N}\left(\mathcal{M}\right)$ und $\mathcal{N}'\left(\mathcal{M}\right)$
      $\mathcal{N} := \mathcal{N}\left(\mathcal{M}\right)$ und $\mathcal{N}' := \mathcal{N}'\left(\mathcal{M}\right)$
    else if $\mathrm{val}\left(C\left[\mathcal{N}\right]\right) - \mathrm{val}\left(C\left[\mathcal{N}'\right]\right) > \sigma_P \cdot \left|\mathrm{val}\left(C\left[\mathcal{N}'\right]\right)\right|$ **then**
      löse $(\mathrm{Proj}[\mathcal{N}, \mathcal{N}'])$
      bestimme $\mathcal{M}$ gemäß Formel (2.2)
      bestimme $\mathcal{N}\left(\mathcal{M}\right)$ und $\mathcal{N}'\left(\mathcal{M}\right)$
      $\mathcal{N} := \mathcal{N}\left(\mathcal{M}\right)$ und $\mathcal{N}' := \mathcal{N}'\left(\mathcal{M}\right)$
    else if $\mathcal{D}\left(\mathcal{N}'\right) > \sigma_D \mathcal{D}_0$ **then**
      wähle $\mathcal{M} \subseteq \mathcal{N}$, so daß die $M' \in \mathcal{N}'|_M$, $M \in \mathcal{M}$, einen hohen Anteil an $\mathcal{D}\left(\mathcal{N}'\right)$ haben
      bestimme $\mathcal{N}'\left(\mathcal{M}\right)$ (ohne Änderung an $\mathcal{N}$)

$\mathcal{N}' := \mathcal{N}'(\mathcal{M})$
end if
until $\mathcal{M} = \emptyset$
Output: die aggregierte Knotenmenge $\mathcal{N}$

2.4 Aktualisierung der konvexen Relaxierung

Ein wichtiger Faktor für den Erfolg des vorgeschlagenen Algorithmusses ist eine Relaxierung $(C[\mathcal{N}])$ von $(P[\mathcal{N}])$ mit kleiner Lücke $|\mathrm{val}(C[\mathcal{N}]) - \mathrm{val}(P[\mathcal{N}])|$. Verbesserungen der Relaxierung können durch zulässige Schnitte erreicht werden, so daß sich die zulässige Menge von $(C[\mathcal{N}])$ verkleinert und sich damit der Optimalwert erhöht. Wir werden hier einen Spaltengenerierungsalgorithmus beschreiben, welcher auf der Beobachtung beruht, daß viele (praxisrelevante) Probleme in kleinere Teilprobleme aufgeteilt werden können, welche dann über ein Master-Problem gekoppelt werden, das sich auch in großer Dimension noch effizient lösen lässt. Schnitte erlauben dann die Einarbeitung von Lösungen der Teilprobleme in das Master-Problem.

Wir beginnen mit einer Einführung in das Konzept der Lagrange-Dekomposition und der Lagrange-Schnitte. Danach wird ein Spaltengenerierungsalgorithmus für die approximative Berechnung der dualen Parameter, die durch die Lagrange-Relaxierung eingeführt wurden, beschrieben. Er hat den Vorteil, daß er nach einer Verfeinerung der Aggregierungen $\mathcal{N}$ oder $\mathcal{N}'$ "warm" gestartet werden kann. Für eine Übersicht zur Lagrange-Dekomposition siehe [17], wo auch ein Spaltengenerierungsalgorithmus im Umfeld von branch-cut-and-price Algorithmen für gemischt-ganzzahlige nichtlineare Programme beschrieben wird.

2.4.1 Lagrange Dekomposition

Sei das Problem (P) block-separabel in der Form, daß eine Partitionierung $\{I_1, \ldots, I_p\}$ existiert, so daß

$$\begin{aligned} F(x) &= \sum_{k=1}^{p} F_k(x_{I_k}), \\ G(x) &= \sum_{k=1}^{p} G_k(x_{I_k}), \\ X &= X_1 \times \cdots \times X_p, \end{aligned}$$

d.h. das Problem (P) liest sich

$$\begin{array}{ll} \min\limits_{x} & \sum\limits_{k=1}^{p} F_k(x_{I_k}) \\ \text{s.d.} & \sum\limits_{k=1}^{p} G_k(x_{I_k}) \le 0 \\ & \qquad x_{I_k} \in X_k \quad k = 1, \ldots, p \end{array} \tag{P}$$

Durch Hinzufügung der koppelnden Restriktionen G, versehen mit dualen Parametern λ, zur Zielfunktion können wir dann (P) in p Teilprobleme kleinerer Größe aufspalten. Die so definierte Lagrangefunktion $L(x;\lambda)$ zerfällt in eine Summe von partiellen Lagrangefunktionen, eine für jeden Block:

$$\begin{aligned} L(x;\lambda) &:= \sum_{k=1}^{p} L_k(x_{I_k};\lambda), \\ L_k(x_{I_k};\lambda) &:= F_k(x_{I_k}) + \lambda^T G_k(x_{I_k}), \qquad k = 1, \ldots, p. \end{aligned}$$

Und das so definierte Lagrangeproblem

$$\begin{array}{ll} \min\limits_{x} & L(x;\lambda) \\ \text{s.d.} & x \in X \end{array} \tag{D_λ}$$

zerfällt in partielle Lagrangeprobleme

$$\begin{array}{ll} \min\limits_{x_{I_k}} & L_k(x_{I_k};\lambda) \\ \text{s.d.} & x_{I_k} \in X_k \end{array} \qquad k = 1,\ldots,p, \tag{$D_{\lambda,k}$}$$

d.h. $\operatorname{val}(D_\lambda) = \sum_{k=1}^{p} \operatorname{val}(D_{\lambda,k})$.

Das duale Problem zu (P) ist nun, einen dualen Parameter $\lambda \geq 0$ zu finden, welcher (D_λ) maximiert:

$$\max_{\lambda \geq 0} \quad \operatorname{val}(D_\lambda) \tag{D}$$

Um eine Beziehung zwischen dem Problem (P) und seinem dualen Problem (D) herzustellen, sei (P) zulässig mit endlichem Optimalwert und x^* ein Optimalpunkt von (P). Dann gilt $G(x^*) \leq 0$ und damit $L(x^*;\lambda) \leq F(x^*) = \operatorname{val}(P)$ für alle $\lambda \geq 0$. Also ist $\operatorname{val}(D_\lambda) \leq L(x^*;\lambda)$ für alle $\lambda \geq 0$, was $\operatorname{val}(D) \leq \operatorname{val}(P)$ impliziert. Die Probleme (D_λ) und (D) liefern deshalb untere Schranken für $\operatorname{val}(P)$. Die Lücke $\operatorname{val}(P) - \operatorname{val}(D)$ wird *Dualitätslücke* genannt.

Um die Eigenschaft der unteren Schranke von (D_λ) auf die konvexe (primale) Relaxierung (C) zu übertragen, kann man nun die Separabilität von (D_λ) zur Generierung von Schnitten nutzen. Angenommen, die Funktionen F_k und G_k sind konvex über $\operatorname{conv}(X_k)$ (also $\check{F} \equiv F$ und $\check{G} \equiv G$), $k = 1,\ldots,p$, so daß die partiellen Lagrangefunktionen $L_k(x_{I_k};\lambda)$ konvex werden für alle $x_{I_k} \in \operatorname{conv}(X_k)$ und alle $\lambda \geq 0$, $k = 1,\ldots,p$. Dann liefert die Restriktion

$$L_k(x_{I_k};\lambda) \geq \operatorname{val}(D_{\lambda,k})$$

einen sogenannten *Lagrange-Schnitt*, eine konvexe Funktion, die zu $\mathcal{C}$ hinzugefügt werden kann. Konstruieren wir diese Schnitte für alle Teilprobleme $k = 1,\ldots,p$ und lösen die Relaxierung (C), wobei wir einen Lösungspunkt $\bar{x}$ erhalten, so gilt $\operatorname{val}(C) \geq \operatorname{val}(D_\lambda)$, denn

$$\begin{aligned} \operatorname{val}(C) &= \check{F}(\bar{x}) = \sum_{k=1}^{p} F_k(\bar{x}_{I_k}) \\ &\geq \sum_{k=1}^{p} F_k(\bar{x}_{I_k}) + \lambda^T G_k(x_{I_k}) \\ &= \sum_{k=1}^{p} L_k(x_{I_k};\lambda) \\ &\geq \sum_{k=1}^{p} \operatorname{val}(D_{\lambda,k}) \\ &= \operatorname{val}(D_\lambda). \end{aligned}$$

Die Abschätzung $\operatorname{val}(P) \geq \operatorname{val}(D) \geq \operatorname{val}(D_\lambda)$ zeigt, daß ein dualer Parameter $\lambda \in \operatorname{Argmin}(D)$ die beste Verbesserung für (C) verspricht, die man mittels Lagrange-Schnitten erreichen kann. Deshalb können duale Lösungsmethoden wie Subgradienten-, duales Schnittebenen- oder Proximal-Bundle-Verfahren, welche die Konkavität der Funktion $\lambda \mapsto \operatorname{val}(D_\lambda)$ und die Konvexität von $\operatorname{feas}(D)$ ausnutzen, genutzt werden um ein optimales λ zu finden. Eine einfachere Methode ist die Verwendung des dualen Lösungspunktes von (C) als eine Approximation von $\lambda \in \operatorname{Argmin}(D)$, was zum sogenannten primalen Schnittebenenverfahren führt. Leider muss dieses Verfahren nicht konvergieren, wie ein Beispiel in [17] zeigt.

2.4.2 Spaltengenerierung

Ein weiteres Verfahren zur Suche nach einem "guten" λ ist die Spaltengenerierungsmethode (column generation method). Hier wird zunächst eine innere Approximation der konvexen Hülle jedes X_k durch die Berechnung von Extrempunkten von X_k berechnet. Bezeichne mit Z_k eine diskrete Teilmenge der Extrempunkte von X_k, so ist $\operatorname{conv}(Z_k) \subseteq \operatorname{conv}(X_k)$. Wir können nun das beschränkte Master-Problem (restricted master problem) formulieren:

$$\begin{array}{ll} \min\limits_{x} & F(x) \\ \text{s.d.} & \begin{array}{l} G(x) \leq 0 \\ \quad x_{I_k} \in \operatorname{conv}(Z_k) \quad k = 1, \ldots, p \end{array} \end{array} \tag{RMP}$$

Wenn (RMP) zulässig ist und $\times_{k=1}^{p} \operatorname{conv}(X_k) \subseteq \check{X}$, so ist $\operatorname{val}(C) \leq \operatorname{val}(\mathrm{RMP})$. Beobachte weiterhin, daß für $\bar{x} \in \operatorname{feas}(\mathrm{RMP})$, d.h. $G(\bar{x}) \leq 0$ und $\bar{x}_{I_k} \in \operatorname{conv}(Z_k)$, $k = 1, \ldots, p$, und für $\lambda \geq 0$

$$\begin{aligned} \operatorname{val}(D_\lambda) &= \sum_{k=1}^{p} \operatorname{val}(D_{\lambda,k}) \\ &= \sum_{k=1}^{p} \min \{L_k(x_{I_k}, \lambda) \,|\, x_{I_k} \in \operatorname{conv}(X_k)\} \\ &\leq \sum_{k=1}^{p} \min \{L_k(x_{I_k}, \lambda) \,|\, x_{I_k} \in \operatorname{conv}(Z_k)\} \\ &\leq \sum_{k=1}^{p} L_k(\bar{x}_{I_k}, \lambda) \\ &\leq \sum_{k=1}^{p} F_k(\bar{x}_{I_k}) \end{aligned}$$

gilt und die Maximierung der linken Seite über $\lambda \geq 0$ und Minimierung der rechten Seite über $\operatorname{feas}(\mathrm{RMP})$ die Ungleichung

$$\operatorname{val}(D) \leq \operatorname{val}(\mathrm{RMP})$$

liefert. Die Vergrößerung von Z_k durch hinzufügen von "guten" Extrempunkten reduziert $\operatorname{val}(\mathrm{RMP})$, während die Berechnung eines "guten" dualen Parameters λ den Wert $\operatorname{val}(D_\lambda)$ erhöht. Spaltengenerierungsmethoden berechnen nun Extrempunkte für Z_k durch Lösung von partiellen Lagrangeproblemen $(D_{\lambda,k})$ unter Benutzung des dualen Parameters den man aus der Lösung von (RMP) erhält. Lösen eines partiellen Problemes liefert unterdessen einen Lagrange-Schnitt für die primale Relaxierung (C). Es kann gezeigt werden, daß dieses Verfahren äquivalent ist zum dualen Schnittebenenverfahren, welches eines polyhedrale Approximation der dualen Funktion $\lambda \mapsto \operatorname{val}(D_\lambda)$ zur Lösung des dualen Problems (D) bestimmt, und welches konvergent ist (vgl. [17]).

Bemerkung. Im Fall linearer Funktionen F und G wird das Problem (RMP) zu einem LP, welches effizient mit einem schnellen LP-Löser gelöst werden kann. Die Lösung eines nichtlinearen nichtkonvexen (RMP) ist viel schwieriger und nicht immer praktikabel. Aber wenn es möglich ist, die Nichtlinearität des Modells in X_k, $k = 1, \ldots, p$, zu "verstecken", so ist die wiederholte Lösung von (RMP) praktikabel, besonders wenn Optimalitätsinformation in aufeinanderfolgenden Lösungsprozessen mit modifizierten Extrempunktmengen wiederverwendet werden kann. Im Fall, daß X_k Ganzzahligkeitsbedingungen an einige Variablen enthält, ist es anderseits vorstellbar, diese Restriktionen nach (RMP) zu übernehmen, um die Qualität der Relaxierung zu verbessern. Im Fall von linearen Funktionen F und G wird das Master-Problem dann ein MIP. Für solche Probleme wurden in den letzten Jahren effiziente Lösungsalgorithmen entwickelt.

Initialisierung von Z_k

Die Initialisierung von (RMP) durch Berechnung von anfänglichen Extrempunkten Z_k, $k = 1, \ldots, p$, scheint eine schwierige Aufgabe zu sein. Ein erster Ansatz besteht darin, mit dem dualen Lösungspunkt λ von (C) die entsprechenden partiellen Lagrangeprobleme $(D_{\lambda,k})$, $k = 1, \ldots, p$, zu lösen. Dies sichert aber noch nicht die Zulässigkeit von (RMP), da die koppelnden Restriktionen $G(\cdot) \leq 0$ i.a. nicht erfüllt sein werden. Daher werden zusätzliche Extrempunkte benötigt, um (RMP) zulässig zu machen. Nowak führte in [17] einen geometrisch motivierten Algorithmus für die Initialisierung von (RMP) ein, aber dieser garantiert auch nicht, daß genug Extrempunkte in jedem Fall gefunden werden.

Da der Sinn des (RMP) die Berechnung einer Approximation des dualen Parameters λ ist, können wir auch die Forderung nach der Zulässigkeit von (RMP) fallen lassen und stattdessen die Verletzung der koppelnden Restriktionen als Strafterm in die Zielfunktion übernehmen:

$$\begin{array}{lll} \min\limits_{x} & F(x) + \delta e^T v_G & \\ & G(x) \leq v_G & \\ \text{s.d.} & x_{I_k} \in \operatorname{conv}(Z_k) & k = 1, \ldots, p \\ & v_G \geq 0 & \end{array} \tag{RMP'}$$

Eine duale Lösung λ von (RMP') kann dann weiterhin als approximierter Lösungspunkt von (D) angesehen werden und wir können damit fortfahren, die partiellen Lagrangeprobleme $(D_{\lambda,k})$ zu formulieren und neue Extrempunkte und Lagrange-Schnitte zu berechnen. Zusätzliche Extrempunkte können die Unzulässigkeit von (RMP) verringern, so daß ein neuer dualer Lösungspunkt von (RMP') eine bessere Approximation des Lösungspunktes von (D) liefern kann.

Eine andere mögliche Vereinfachung besteht in der Abschwächung der Forderung $Z_k \subseteq X_k$ zu $Z_k \subseteq \operatorname{conv}(X_k)$. Dies hat keinen weiteren Einfluss auf die Qualität der inneren Approximation, da $\operatorname{val}(D_{\lambda,k}) = \min\{L_k(x_{I_k}; \lambda) \,|\, x_{I_k} \in \operatorname{conv}(X_k)\}$, wie wir oben schon gezeigt haben.

Erforschte Teilprobleme

Wenn die konvexe Relaxierung mittels Lagrange-Schnitten aktualisiert wird, so ist es ausreichend, nur die partiellen Lagrangeprobleme $(D_{\lambda,k})$, $k = 1, \ldots, p$, zu lösen, welche nicht bereits "erforscht" sind, d.h. deren Lücke zwischen innerer und äußerer Approximation von $\operatorname{conv}(X_k)$ weiterhin zu groß ist: Die konvexe Relaxierung (äußere Approximation) von $(D_{\lambda,k})$ ist

$$\begin{array}{lll} \min\limits_{x_{I_k}} & L_k(x_{I_k}; \lambda) & \\ \text{s.d.} & x_{I_k} \in \check{X}_k & \\ & c(x_{I_k}) \leq 0 & c \in \mathcal{C}_k \end{array} \tag{$C_{\lambda,k}$}$$

wobei $\check{X}_k$ eine konvexe Obermenge von X_k ist und $\mathcal{C}_k$ die Schnitte von $\mathcal{C}$ bezeichne, die nur auf den k-ten Block wirken. Betrachte weiterhin die innere Approximation von $\operatorname{conv}(D_{\lambda,k}) := \min_{x_{I_k}} \{L_k(x_{I_k}; \lambda) \,|\, x_{I_k} \in \operatorname{conv}(X_k)\}$, die man mittels Ersetzung von $\operatorname{conv}(X_k)$ durch $\operatorname{conv}(Z_k)$ erhält, d.h.

$$\begin{array}{ll} \min\limits_{x_{I_k}} & L_k(x_{I_k}; \lambda) \\ \text{s.d.} & x_{I_k} \in \operatorname{conv}(Z_k) \end{array} \tag{$\mathrm{RMP}_{\lambda,k}$}$$

Damit lässt sich der Optimalwert von $(D_{\lambda,k})$ von unten und nach oben durch die Ungleichung

$$\operatorname{val}(C_{\lambda,k}) \leq \operatorname{val}(D_{\lambda,k}) \leq \operatorname{val}(\mathrm{RMP}_{\lambda,k})$$

beschränken. Wenn die Lücke $\operatorname{val}(\mathrm{RMP}_{\lambda,k}) - \operatorname{val}(C_{\lambda,k})$ klein ist, so muss das k-te partielle Lagrangeproblem nicht gelöst werden, da ein zusätzlicher Lagrange-Schnitt und ein zusätzlicher Extrempunkt die Relaxierungen nicht wesentlich verbessern können.

2.4.3 Übertragung auf das aggregierte Problem

Nach der Einführung der Lagrange-Dekomposition und Spaltengenerierung für den allgemeinen Fall oben, fahren wir mit der Notation für die Aggregierung im diskreten Fall fort und werfen einen Blick auf den Warmstart der Spaltengenerierung nach einer Verfeinerung der Aggregierung.

Angenommen, die Daten von $(P[\mathcal{N}])$ haben folgende block-separable Gestalt:

$$\begin{aligned}
F_{\mathcal{N}}(x) &= \sum_{k=1}^{p} F_{k,\mathcal{N}}(x_{I_k}), \\
G_{\mathcal{N}}(x) &= \sum_{k=1}^{p} G_{k,\mathcal{N}}(x_{I_k}), \\
X_{\mathcal{N}} &= X_{1,\mathcal{N}} \times \cdots \times X_{p,\mathcal{N}}.
\end{aligned}$$

Dann erhalten wir das aggregierte Problem

$$\begin{array}{ll}
\min\limits_{x} & \sum_{k=1}^{p} F_{k,\mathcal{N}}(x_{I_k}) \\
\text{s.d.} & \sum_{k=1}^{p} G_{k,\mathcal{N}}(x_{I_k}) \leq 0 \\
& \qquad x_{I_k} \in X_{k,\mathcal{N}} \quad k = 1, \ldots, p
\end{array} \tag{$P[\mathcal{N}]$}$$

wobei $x_{I_k} = (x_{I_k,N})_{N \in \mathcal{N}}$.

Das Problem $(P[\mathcal{N}])$ zerfällt nun in p Teilprobleme durch Dualisierung der koppelnden Restriktionen $G_{\mathcal{N}}$: Wir führen die partielle Lagrangefunktion

$$L_{k,\mathcal{N}}(x_{I_k}; \lambda) := F_{k,\mathcal{N}}(x_{I_k}) + \lambda^T G_{k,\mathcal{N}}(x_{I_k})$$

ein und erhalten die partiellen Lagrangeprobleme

$$\begin{array}{ll}
\min\limits_{x_{I_k}} & L_{k,\mathcal{N}}(x_{I_k}; \lambda) \\
\text{s.d.} & x_{I_k} \in X_{k,\mathcal{N}}
\end{array} \tag{$D_{\lambda,k}[\mathcal{N}]$}$$

Zusammenfügung der partiellen Lagrangeprobleme ergibt das aggregierte duale Problem

$$\max_{\lambda \geq 0} \quad \sum_{k=1}^{p} \operatorname{val}(D_{\lambda,k}[\mathcal{N}]) \tag{$D[\mathcal{N}]$}$$

Lagrange-Schnitte

Lösung eines partiellen Lagrangeproblemes für einen dualen Parameter $\lambda \geq 0$ liefert einen Lagrange-Schnitt

$$L_{k,\mathcal{N}}(x_{I_k}; \lambda) \geq \operatorname{val}(D_{\lambda,k}[\mathcal{N}]),$$

welcher zur Aktualisierung der konvexen Relaxierung $(C[\mathcal{N}])$ von $(P[\mathcal{N}])$ genutzt werden kann, d.h. die Funktion $\operatorname{val}(D_{\lambda,k}[\mathcal{N}]) - L_{k,\mathcal{N}}(\cdot_{I_k}; \lambda)$ wird zu $\mathcal{C}_{\mathcal{N}}$ hinzugefügt.

Die Lösung aller partiellen Lagrangeprobleme und Hinzufügung der erhaltenen Schnitte zu $\mathcal{C}_{\mathcal{N}}$ liefert die Abschätzung $\operatorname{val}(C[\mathcal{N}]) \geq \sum_{k=1}^{p} \operatorname{val}(D_{\lambda,k}[\mathcal{N}])$, so daß die "beste" Aktualisierung von $(C[\mathcal{N}])$ durch ein $\lambda \geq 0$ erreicht wird, welches $(D[\mathcal{N}])$ löst. Da das Lösen von $(D[\mathcal{N}])$ teuer sein kann, benutzen wir das Spaltengenerierungsverfahren um eine approximative Lösung von $(D[\mathcal{N}])$ zu bestimmen.

Spaltengenerierung

Bezeichne mit $Z_{k,\mathcal{N}}$ eine diskrete Menge von Extrempunkten von $X_{k,\mathcal{N}}$, so daß $\operatorname{conv}(Z_{k,\mathcal{N}}) \subseteq \operatorname{conv}(X_{k,\mathcal{N}})$. Das aggregierte beschränkte Master-Problem lässt sich formulieren als

$$\begin{array}{ll} \min\limits_{x} & F_{\mathcal{N}}(x) \\ \text{s.d.} & \begin{array}{l} G_{\mathcal{N}}(x) \leq 0 \\ \quad x_{I_k} \in \operatorname{conv}(Z_{k,\mathcal{N}}) \quad k = 1, \ldots, p \end{array} \end{array} \tag{RMP[$\mathcal{N}$]}$$

Für (RMP[$\mathcal{N}$]) gilt die Abschätzung $\operatorname{val}(D[\mathcal{N}]) \leq \operatorname{val}(\text{RMP}[\mathcal{N}])$ und diese Lücke lässt sich durch Hinzufügung von Extrempunkten zu $Z_{k,\mathcal{N}}$ reduzieren. Die Berechnung von Extrempunkten durch Lösen von partiellen Lagrangeproblemen liefert zusätzlich Lagrange-Schnitte, welche zu $\mathcal{C}_{\mathcal{N}}$ hinzugefügt werden. Um die Anzahl der in jeder Iteration zu lösenden Teilprobleme zu reduzieren, betrachte die äußere Approximation

$$\begin{array}{ll} \min\limits_{x_{I_k}} & L_{k,\mathcal{N}}(x_{I_k}; \lambda) \\ \text{s.d.} & \begin{array}{ll} \quad x_{I_k} \in \check{X}_{k,\mathcal{N}} & \\ c(x_{I_k}) \leq 0 & c \in \mathcal{C}_{k,\mathcal{N}} \end{array} \end{array} \tag{$C_{\lambda,k}[\mathcal{N}]$}$$

und innere Approximation

$$\begin{array}{ll} \min\limits_{x_{I_k}} & L_{k,\mathcal{N}}(x_{I_k}; \lambda) \\ \text{s.d.} & x_{I_k} \in \operatorname{conv}(Z_{k,\mathcal{N}}) \end{array} \tag{$\text{RMP}_{\lambda,k}[\mathcal{N}]$}$$

von $\operatorname{conv}(D_{\lambda,k}[\mathcal{N}])$ bzgl. einer konvexen Obermenge $\check{X}_{k,\mathcal{N}}$ von $X_{k,\mathcal{N}}$ und den Schnitten $\mathcal{C}_{k,\mathcal{N}}$ aus $\mathcal{C}_{\mathcal{N}}$, die sich nur auf Block k beziehen, $k = 1, \ldots, p$.

Verfeinerungsoperationen

Im Verfeinerungsalgorithmus aus Abschnitt 2.3 wird eine Aggregierung $\mathcal{N}$ iterativ verfeinert. Nach jeder Verfeinerungsoperation müssen die konvexen Relaxierungen ausgewertet werden. Da das Problem $(P[\mathcal{N}'])$ für eine Verfeinerung $\mathcal{N}'$ von $\mathcal{N}$ ähnlich zu $(P[\mathcal{N}])$ sein kann, ist es wünschenswert, die Schnitte $\mathcal{C}_{\mathcal{N}}$ von $(C[\mathcal{N}])$ nach $\mathcal{C}_{\mathcal{N}'}$ zu übernehmen. Weiterhin, wenn man die Spaltengenerierung und die Schwierigkeiten mit ihrer Initialisierung berücksichtigt, so wäre es vorzuziehen, die Extrempunkte $Z_{k,\mathcal{N}}$ zu Extrempunkten $Z_{k,\mathcal{N}'}$ zu aktualisieren, um eine innere Approximation von $\operatorname{conv}(X_{k,\mathcal{N}'})$ zu bilden, $k = 1, \ldots, p$.

Wir nehmen daher an, daß ein Punkt $x_{I_k} = (x_{I_k,N})_{N \in \mathcal{N}} \in X_{k,\mathcal{N}}$ als Punkt $x'_{I_k} = \left(x'_{I_k,N'}\right)_{N' \in \mathcal{N}'} \in X_{k,\mathcal{N}'}$ dargestellt werden kann, $k = 1, \ldots, p$. Diese Operation wird durch Operatoren $E_k : X_{k,\mathcal{N}} \to X_{k,\mathcal{N}'}$ ausgedrückt. Wir erhalten damit

$$E_k(X_{k,\mathcal{N}}) \subseteq X_{k,\mathcal{N}'}.$$

Betrachte nun einen Lagrange-Schnitt $L_{k,\mathcal{N}}(\cdot_{I_k}; \lambda) \geq \operatorname{val}(D_{\lambda,k}[\mathcal{N}])$ aus $\mathcal{C}_{\mathcal{N}}$. Für einen Punkt $x^*_{I_k} \in \operatorname{Argmin}(D_{\lambda,k}[\mathcal{N}])$ muss $E_k\left(x^*_{I_k}\right)$ nicht optimal für $(D_{\lambda,k}[\mathcal{N}'])$ sein, so daß wir den Schnitt $L_{k,\mathcal{N}'}(\cdot_{I_k}; \lambda) \geq L_{k,\mathcal{N}'}\left(E_k\left(x^*_{I_k}\right); \lambda\right)$ nicht einfach zu $\mathcal{C}_{\mathcal{N}'}$ hinzufügen können. Aber angenommen, die Probleme $(D_{\lambda,k}[\mathcal{N}])$ und $(D_{\lambda,k}[\mathcal{N}'])$ sind ähnlich, so ist die Lösung von $(D_{\lambda,k}[\mathcal{N}'])$ mit einem Warmstart ausgehend von $E_k\left(x^*_{I_k}\right)$ vertretbar. Der Schnitt $L_{k,\mathcal{N}'}(\cdot_{I_k}; \lambda) \geq \operatorname{val}(D_{\lambda,k}[\mathcal{N}'])$ kann dann zu $\mathcal{C}_{\mathcal{N}'}$ hinzugefügt werden.

Für die Extrempunkte $Z_{k,\mathcal{N}}$ beobachten wir, daß es nicht nötig ist, daß sie Optimalpunkte eines Problemes $(D_{\lambda,k}[\mathcal{N}])$ für einen dualen Parameter $\lambda \geq 0$ sind. Wir können einfachen $Z_{k,\mathcal{N}'} := E_k(Z_{k,\mathcal{N}'})$ setzen, so daß dann $Z_{k,\mathcal{N}'} \subset X_{k,\mathcal{N}'}$ gilt. Doch die Punkte $Z_{k,\mathcal{N}'}$ können verbessert werden, ähnlich wie bei der Aktualisierung der Lagrange-Schnitte, durch Lösung partieller Lagrangeprobleme mit Warmstart ausgehend von Punkten aus $Z_{k,\mathcal{N}'}$.

2.5 Adaptiver Diskretisierungsalgorithmus

Wir sind nun bereit für eine abstrakte Formulierung des adaptiven Diskretisierungsalgorithmusses:

Algorithmus 2.2 (adaptive Diskretisierung).
Input: initiale Knotenmenge $\bar{N}$, Parameter $\bar{m} \in \mathbb{N}$ und $\sigma_P \in [0,1]$, $\sigma_D \in [0,1]$
Compute:
berechne anfängliche grobe Aggregierungen $\mathcal{N}$ und $\mathcal{N}'$ von $\bar{N}$, $\mathcal{N}' \preceq \mathcal{N}$
$\mathcal{D}_0 := \mathcal{D}(\mathcal{N}')$
initialisiere (RMP[$\mathcal{N}$]) durch Bestimmung anfänglicher Extrempunkte $Z_{k,\mathcal{N}} \subseteq X_{k,\mathcal{N}}$
repeat
aktualisiere die konvexe Relaxierung ($C[\mathcal{N}]$) durch Schnittgenerierung, siehe Algorithmus 2.3
berechne $\mathcal{C}_{\mathcal{N}'}$ aus $\mathcal{C}_{\mathcal{N}}$
löse ($C[\mathcal{N}]$) und ($C[\mathcal{N}']$)
if $\mathrm{val}(C[\mathcal{N}']) - \mathrm{val}(C[\mathcal{N}]) > \sigma_P \cdot |\mathrm{val}(C[\mathcal{N}])|$ **then**
berechne $\mathcal{M}$ entsprechend Formel (2.1)
berechne $\mathcal{N}(\mathcal{M})$ und $\mathcal{N}'(\mathcal{M})$
$Z_{k,\mathcal{N}(\mathcal{M})} := E_k(Z_{k,\mathcal{N}})$, $k = 1, \ldots, p$.
$\mathcal{N} := \mathcal{N}(\mathcal{M})$ und $\mathcal{N}' := \mathcal{N}'(\mathcal{M})$
else if $\mathrm{val}(C[\mathcal{N}]) - \mathrm{val}(C[\mathcal{N}']) > \sigma_P \cdot |\mathrm{val}(C[\mathcal{N}'])|$ **then**
löse (Proj[$\mathcal{N},\mathcal{N}'$])
berechne $\mathcal{M}$ entsprechend Formel (2.2)
berechne $\mathcal{N}(\mathcal{M})$ und $\mathcal{N}'(\mathcal{M})$
$Z_{k,\mathcal{N}(\mathcal{M})} := E_k(Z_{k,\mathcal{N}})$, $k = 1, \ldots, p$.
$\mathcal{N} := \mathcal{N}(\mathcal{M})$ und $\mathcal{N}' := \mathcal{N}'(\mathcal{M})$
else if $\mathcal{D}(\mathcal{N}') > \sigma_D \mathcal{D}_0$ **then**
wähle $\mathcal{M} \subseteq \mathcal{N}$, so daß die $M' \in \mathcal{N}'|_M$, $M \in \mathcal{M}$, einen hohen Anteil an $\mathcal{D}(\mathcal{N}')$ haben
berechne $\mathcal{N}'(\mathcal{M})$ (ohne Änderung an $\mathcal{N}$)
$\mathcal{N}' := \mathcal{N}'(\mathcal{M})$
end if
until $\mathcal{M} = \emptyset$
Output: die aggregierte Knotenmenge $\mathcal{N}$

Die Spaltengenerierungsmethode zur Generierung von Schnitten $\mathcal{C}_{\mathcal{N}}$ ist im folgenden Algorithmus dargestellt. Ein Teilproblem ($D_{\lambda,k}[\mathcal{N}]$) wird als "erforscht" angesehen, wenn der relative Abstand zwischen den Optimalwerten der inneren Approximation ($\mathrm{RMP}_{\lambda,k}[\mathcal{N}]$) und der äußeren Approximation ($C_{\lambda,k}[\mathcal{N}]$) von $\mathrm{conv}(D_{\lambda,k}[\mathcal{N}])$ den Parameter σ_E nicht überschreitet.

Algorithmus 2.3 (Spaltengenerierung).
Input: Extrempunkte $Z_{k,\mathcal{N}}$, $k = 1, \ldots, p$, und eine Toleranz $\sigma_E > 0$
Compute:
löse (RMP[$\mathcal{N}$]), um einen dualen Optimalpunkt λ zu erhalten
repeat
for all $k = 1, \ldots, p$ **do**
löse ($C_{\lambda,k}[\mathcal{N}]$) und erhalte einen primalen Optimalpunkt $\hat{x}_{I_k}$
if $\frac{\mathrm{val}(\mathrm{RMP}_{\lambda,k}[\mathcal{N}]) - \mathrm{val}(C_{\lambda,k}[\mathcal{N}])}{|\mathrm{val}(C_{\lambda,k}[\mathcal{N}])|} > \sigma_E$ **then**
berechne einen Optimalpunkt z_{I_k} von ($D_{\lambda,k}[\mathcal{N}]$) unter Benutzung von $\hat{x}_{I_k}$ als Startpunkt
füge z_{I_k} zu $Z_{k,\mathcal{N}}$ hinzu und den Schnitt $-L_{k,\mathcal{N}}(\cdot_{I_k}; \lambda) + \mathrm{val}(D_{\lambda,k}[\mathcal{N}])$ zu $\mathcal{C}_{\mathcal{N}}$
end if
end for
löse (RMP[$\mathcal{N}$]), um neue primale und duale Lösungen x^* und λ zu erhalten
entferne Extrempunkte aus $Z_{k,\mathcal{N}}$ mit verschwindenden Konvexkombinationsparameter in x^*
until $\mathrm{val}(\mathrm{RMP}[\mathcal{N}])$ ändert sich nicht und $Z_{k,\mathcal{N}}$ wurde für kein $k \in \{1, \ldots, p\}$ vergrößert
Output: eine aktualisierte Menge von Extrempunkten $Z_{k,\mathcal{N}}$, $k = 1, \ldots, p$, und Schnitte $\mathcal{C}_{\mathcal{N}}$

Bemerkung. Natürlich sind viele Variationen von Algorithmus 2.3 vorstellbar. Variation zwischen einem zulässigen und unzulässigen Master-Problem eröffnet viele Möglichkeiten, um mit Unzulässigkeiten umzugehen. Auch die Wahl der Teilprobleme, die gelöst werden sollen, ist eine knifflige Aufgabe. Zum Beispiel, anstatt alle partiellen Lagrangeprobleme zu lösen, deren relativer Abstand zwischen $\mathrm{val}(C_{\lambda,k}[\mathcal{N}])$ und $\mathrm{val}(\mathrm{RMP}_{\lambda,k}[\mathcal{N}])$ den Wert σ_E überschreitet, könnte man zunächst die Teilprobleme lösen, bei denen der relative Abstand am größten ist, und danach $(\mathrm{RMP}[\mathcal{N}])$ nochmals lösen, um den dualen Parameter zu aktualisieren.

Kapitel 3

Ein adaptiver Diskretisierungsalgorithmus für mehrstufige stochastische Programme

In diesem Kapitel wird der abstrakte Algorithmus aus dem letztem Kapitel auf die Diskretisierung mehrstufiger stochastischer Programme übertragen. Zunächst führen wir etwas theoretischen Hintergrund über konvexe zweistufige stochastische Programme ein und leiten daraus Szenarioreduktionsalgorithmen ab ([7, 15]). Weiterhin formulieren wir die Szenariobaumform eines MSPs und beschreiben unseren Ansatz zur Aggregierung und Verfeinerung des Szenariobaumes. Das nächste Thema wird eine Übersicht über Dekompositionsverfahren für mehrstufige stochastische Programme sein. Danach sind wir bereit, den adaptiven Szenarioreduktionsalgorithmus in seiner letztendlichen Form anzugeben. Dieses Kapitel endet mit einer Betrachtung des Spezialfalles gemischt-ganzzahliger linearer MSPs.

3.1 Deterministisches Äquivalent, Stabilität, Szenarioreduktion

3.1.1 Deterministisches Äquivalent

Wir erinnern uns an das mehrstufige stochastische Programm (MSP), Seite 8. An dieser Stelle geben wir nur eine kurze Einführung in das Verfahren der dynamischen Programmierung, welches uns zum deterministischen Äquivalent von (MSP) führen wird, für mehr Details siehe [19]. Wir beginnen mit der Definition der zugrundeliegenden Rekursionsformeln:

Für die letzte Zeitstufe, $t = T$, definiere die Funktion $f_T : \Omega \times \mathbb{R}^r \to \bar{\mathbb{R}} := \mathbb{R} \cup \{\pm\infty\}$ mittels

$$f_T(\omega, x_1, \ldots, x_T) := \begin{cases} f(x_1, \ldots, x_T, \xi(\omega)) & x_t \in X_t(\zeta_t), \; g_t(x_1, \ldots, x_t, \zeta_t(\omega)) \leq 0, \; t = 1, \ldots, T \\ +\infty & \text{sonst} \end{cases}$$

für alle $\omega \in \Omega$, $x_t \in \mathbb{R}^{r_t}$, $t = 1, \ldots, T$. Seien weiterhin f, g_t und X_t, $t = 1, \ldots, T$, so gegeben, daß Funktionen $\bar{f}_T : \Omega \times \mathbb{R}^r \to \bar{\mathbb{R}}$ und $q \in L^1(\Omega, \mathcal{F}, \mathbb{P})$ existieren, die die folgenden Eigenschaften besitzen:

- $\bar{f}_T$ ist ein normaler Integrand: Die mengenwertige Abbildung $\omega \mapsto \mathrm{epi}\bar{f}_T(\omega,\cdot) := \{(x,\alpha) \in \mathbb{R}^r \times \mathbb{R} \,|\, \bar{f}_T(\omega,x) \leq \alpha\}$ ist messbar $(\{\omega \in \Omega \,|\, \exists\,(x,\alpha) \in O : \bar{f}_T(x,\omega) \leq \alpha\} \in \mathcal{F}$ für alle offenen Mengen $O \subseteq \mathbb{R}^r \times \mathbb{R}$) und abgeschlossen-wertig.
- $\{x \in \mathbb{R}^r \,|\, \bar{f}_T(\omega,x) \leq \alpha\}$ ist kompakt für alle $\omega \in \Omega$ und alle $\alpha \in \mathbb{R}$.
- $\mathbb{P}\left[f_T(\omega,x) = \bar{f}_T(\omega,x)\ \forall x \in \mathbb{R}^r\right] = 1$.
- $\forall x \in \mathbb{R}^r$: $f_T(\omega,x) \geq q(\omega)$ $\mathbb{P}$-f.s.

Jetzt können wir rekursiv die folgenden Funktionen definieren ("*Bellmann-Rekursion*"):

$$\varphi_t(\omega, x_1(\omega), \ldots, x_t(\omega)) := \mathbb{E}\left[f_t(\cdot, x_1(\cdot), \ldots, x_t(\cdot))\,|\mathcal{F}_t\right](\omega) \qquad \mathbb{P}\text{-f.s.}, \quad t = T, \ldots, 1, \tag{3.1}$$

$$f_{t-1}(\omega, x_1, \ldots, x_{t-1}) := \inf_{y \in \mathbb{R}^{r_t}} \varphi_t(\omega, x_1, \ldots, x_{t-1}, y) \qquad \mathbb{P}\text{-f.s.}, \quad t = T, \ldots, 1. \tag{3.2}$$

Man kann folgenden Satz beweisen ([10, Satz 1 und 2], [19]):

Theorem 3.1. *Seien die oberen Bedingungen an f_T erfüllt und existiere eine Funktion $\tilde{x} : \Omega \to \mathbb{R}^r$ mit $\tilde{x}_t$ $\mathcal{F}_t$-messbar, $t = 1, \ldots, T$, und $\mathbb{E}\left[f_T(\omega, \tilde{x}_1(\omega), \ldots, \tilde{x}_T(\omega))\right] < \infty$. Dann ist die Abbildung $\bar{x} : \Omega \to \mathbb{R}^r$ genau dann optimal für (MSP), wenn die Bellmann-Gleichungen*

$$\varphi_t(\omega, \bar{x}_1(\omega), \ldots, \bar{x}_t(\omega)) = f_{t-1}(\omega, \bar{x}_1(\omega), \ldots, \bar{x}_{t-1}(\omega)) \qquad \mathbb{P}\text{-f.s.},\ t = 1, \ldots, T \tag{3.3}$$

erfüllt sind. Weiterhin existiert eine optimale Lösung von (MSP).

Für $t = 1$ liefern die Formeln der Bellmann-Rekursion (3.1), (3.2) das deterministische Optimierungsproblem

$$f_0 := \inf_{x_1 \in \mathbb{R}^{r_1}} \mathbb{E}\left[f_1(\omega, x_1)\right] \tag{DA}$$

zur Bestimmung einer optimalen Erststufenentscheidung. Es wird mit *deterministisches Äquivalent* von (MSP) bezeichnet. Satz 3.1 gibt Bedingungen an, unter denen (DA) lösbar ist und wann eine Lösung $\bar{x}_1$ von (DA) zu einer Lösung $\bar{x}(\cdot)$ von (MSP) ausgebaut werden kann.

3.1.2 Stabilität und Wahrscheinlichkeitsmetriken

Wenn man mehrstufige stochastische Programme lösen möchte, so ist man mit dem zugrundeliegenden Wahrscheinlichkeitsmass konfrontiert. Das Wahrscheinlichkeitsmass $\mathbb{P}$ muss nicht exakt bekannt sein oder muss keinen endlichen Träger besitzen. Auch im Falle endlich vieler Szenarien ist das Problem (MSP) oft sehr groß. Andererseits bedeutet das Lösen des deterministischen Äquivalents (DA) die Behandlung einer Zielfunktion, die nur implizit gegeben ist.

Um diese Probleme zu behandeln, wurden viele Approximationsverfahren entwickelt, welche $\mathbb{P}$ durch ein diskretes Wahrscheinlichkeitsmass $\mathbb{Q}$ mit kleinerem Träger ersetzen und die Eigenschaft haben, daß der dazugehörige Optimalwert möglichst nahe bei $\mathrm{val}(\mathrm{MSP}) = \mathrm{val}(\mathrm{DA})$ bleibt. In diesem Kontext werden Stabilitätseigenschaften bzgl. Approximationen von $\mathbb{P}$ wichtig. Leider sind solche Eigenschaften für den mehrstufigen Fall ($T > 2$) bisher nicht bekannt. Wir präsentieren hier einige Ergebnisse aus [7] über die Stabilität von konvexen zweistufigen stochastischen Programmen. Diese Überlegungen führen zu Szenarioreduktionsalgorithmen, die wir danach beschreiben werden.

Ein zweistufiges stochastisches Programm kann ähnlich zu (DA) formuliert werden als

$$\inf_{x \in X} \mathbb{E}_{\mathbb{P}}\left[f(\cdot, x)\right] = \inf_{x \in X} \int_{\Omega} f(\omega, x)\, \mathbb{P}[d\omega] \tag{3.4}$$

wobei die konvexe Menge $X \subseteq \mathbb{R}^r$ die Menge der zulässigen Erststufenentscheidungen beschreibt und die Funktion $f : \Omega \times \mathbb{R}^r \to \bar{\mathbb{R}}$ eine ähnliche Form hat wie die Funktion f_{T-1} in (3.2), d.h.

$$f(\omega, x) = \inf \left\{ \tilde{f}(\omega, x, y) \, | y \in Y(\xi(\omega)), g(x, y, \xi(\omega)) \leq 0 \right\}$$

für eine Funktion $\tilde{f} : \Omega \times \mathbb{R}^r \times \mathbb{R}^{r_2} \to \bar{\mathbb{R}}$. Weiterhin nehmen wir an, daß die Funktion f messbar bzgl. ω und unterhalbstetig und konvex bzgl. x ist. Sei $\mathbb{E}_{\mathbb{P}}[\cdot]$ der Erwartungswert bzgl. des Wahrscheinlichkeitsmasses $\mathbb{P}$ auf $(\Omega, \mathcal{F})$. Wir bezeichnen den Raum der Wahrscheinlichkeitsmaße auf $(\Omega, \mathcal{F})$ mit $\mathcal{P}(\Omega)$.

Für $\mathbb{Q} \in \mathcal{P}(\Omega)$ sei

$$v(\mathbb{Q}) := \inf_{x \in \mathbb{R}^r} \mathbb{E}_{\mathbb{Q}}[f(\cdot, x)]$$

der Optimalwert von (3.4) bzgl. des Wahrscheinlichkeitsmaßes $\mathbb{Q}$. Die Menge der ε-approximativen Lösungen bezeichnen wir mit

$$S_\varepsilon(\mathbb{Q}) := \{x \in \mathbb{R}^r \, | \mathbb{E}_{\mathbb{Q}}[f(\cdot, x)] \leq v(\mathbb{Q}) + \varepsilon\}$$

für jedes $\varepsilon \geq 0$ und sei $S(\mathbb{Q}) := S_0(\mathbb{Q})$ die Menge der Lösungspunkte.

Man betrachte nun die Menge

$$\mathcal{P}_f(\Omega) := \left\{ \mathbb{Q} \in \mathcal{P}(\Omega) \middle| -\infty < \int_\Omega \inf_{\|x\| \leq \rho} f(\omega, x) \, \mathbb{Q}[d\omega] \leq \sup_{\|x\| \leq \rho} \int_\Omega f(\omega, x) \, \mathbb{Q}[d\omega] < \infty \quad \forall \rho > 0 \right\}$$

von Wahrscheinlichkeitsmaßen und den Wahrscheinlichkeitsabstand

$$d_{f,\rho}(\mathbb{P}, \mathbb{Q}) := \sup_{\|x\| \leq \rho} |\mathbb{E}_{\mathbb{P}}[f(\cdot, x)] - \mathbb{E}_{\mathbb{Q}}[f(\cdot, x)]|$$

für jedes $\rho > 0$ und $\mathbb{P}, \mathbb{Q} \in \mathcal{P}_f(\Omega)$. Um die Variation von $v(\cdot)$ und $S_\varepsilon(\cdot)$ durch Veränderung des Wahrscheinlichkeitsmaßes abzuschätzen, definieren wir die Wachstumsfunktion ψ von (3.4) durch

$$\psi(\tau) := \min_{x: \, d(x, S(\mathbb{P})) \geq \tau} \int_\Omega f(\omega, x) \, \mathbb{Q}[d\omega] - v(\mathbb{P})$$

für $\tau \geq 0$, und $d(x, C) := \inf_{y \in C} \|x - y\|$ dem Abstand von $x \in \mathbb{R}^r$ zu $C \subseteq \mathbb{R}^r$. Weiterhin sei $\Psi(\eta) := \eta + \psi^{-1}(2\eta)$ für $\eta \geq 0$ und $\mathbb{B} := \{x \in \mathbb{R}^r \, | \|x\| \leq 1\}$ die abgeschlossene Einheitskugel im $\mathbb{R}^r$. Letztendlich bezeichne

$$\mathbf{d}_\infty(C, D) := \sup_{x \in \mathbb{R}^r} |d(x, C) - d(x, D)|$$

den Pompeiu-Hausdorff Abstand zweier nichtleerer abgeschlossener Mengen $C, D \subseteq \mathbb{R}^r$. Dann kann man folgendes Stabilitätsresultat für den Optimalwert und die ε-approximativen Lösungsmengen zeigen:

Theorem 3.2. *Sei $\mathbb{P} \in \mathcal{P}_f(\Omega)$ und $S(\mathbb{P})$ nichtleer und beschränkt. Dann existieren Konstanten $\rho > 0$ und $\bar{\varepsilon} > 0$, so daß*

$$|v(\mathbb{P}) - v(\mathbb{Q})| \leq d_{f,\rho}(\mathbb{P}, \mathbb{Q}) \quad \textit{und} \quad \emptyset \neq S(\mathbb{Q}) \subseteq S(\mathbb{P}) + \Psi(d_{f,\rho}(\mathbb{P}, \mathbb{Q})) \, \mathbb{B}$$

für alle $\mathbb{Q} \in \mathcal{P}_f(\Omega)$ mit $d_{f,\rho}(\mathbb{P}, \mathbb{Q}) < \bar{\varepsilon}$, und es gilt für jedes $\varepsilon \in (0, \bar{\varepsilon})$

$$\mathbf{d}_\infty(S_\varepsilon(\mathbb{P}), S_\varepsilon(\mathbb{Q})) \leq \frac{2\rho}{\varepsilon} d_{f,\rho+\varepsilon}(\mathbb{P}, \mathbb{Q})$$

für alle $\mathbb{Q} \in \mathcal{P}_f(\Omega)$ mit $d_{f,\rho}(\mathbb{P}, \mathbb{Q}) < \varepsilon$.

Während es schwer ist, den Wahrscheinlichkeitsabstand $d_{f,\rho}$ explizit zu berechnen, kann man ihn aber nach oben mit einer Wahrscheinlichkeitsmetrik abschätzen, welche die Funktion f nur indirekt einbezieht. Für das Wahrscheinlichkeitsmass $\mathbb{Q}$, berechnet von einem Szenarioreduktionsalgorithmus, lässt sich die Variation der Optimalwerte dann mittels Satz 3.2 kontrollieren.

Der Integrand $f(\cdot, x)$ von (3.4) ist oft nicht differenzierbar, aber lokal Lipschitz-stetig auf Ω. In vielen Fällen existiert eine messbare stetige Metrik $c : \Omega \times \Omega \to \mathbb{R}^+$ und eine nichtfallende Funktion $g : \mathbb{R}^+ \to \mathbb{R}^+ \setminus \{0\}$, so daß für alle $x \in \mathbb{R}^r$ und $\omega, \tilde{\omega} \in \Omega$

$$|f(\omega, x) - f(\tilde{\omega}, x)| \leq g(\|x\|)\, c(\omega, \tilde{\omega}), \tag{3.5}$$

es ist also $f(\cdot, x) \in F_{g(\|x\|)c}$ wobei

$$F_c := \{ f : \Omega \to \mathbb{R} \,|\, f(\omega) - f(\tilde{\omega}) \leq c(\omega, \tilde{\omega}) \quad \forall \omega, \tilde{\omega} \in \Omega \}.$$

Für die Metrik c existiert (aufgrund der Dreiecksungleichung) eine messbare Funktion $\lambda : \Omega \to \mathbb{R}^+$, welche auf beschränkten Mengen beschränkt ist und die Eigenschaft $c(\omega, \tilde{\omega}) \leq \lambda(\omega) + \lambda(\tilde{\omega})$ für alle $\omega, \tilde{\omega} \in \Omega$ hat.

Wir betrachten nun die Wahrscheinlichkeits-(semi-)metrik[1]

$$\zeta_c(\mathbb{P}, \mathbb{Q}) := \sup_{f \in F_c} \left| \int_\Omega f(\omega)\, \mathbb{P}[d\omega] - \int_\Omega f(\omega)\, \mathbb{Q}[d\omega] \right|$$

und das sogenannte *Kantorovich-Funktional*

$$\hat{\mu}_c(\mathbb{P}, \mathbb{Q}) := \inf \left\{ \int_{\Omega \times \Omega} c(\omega, \tilde{\omega})\, \eta(d(\omega, \tilde{\omega})) \middle| \eta \in \mathcal{P}(\Omega \times \Omega),\, \forall B \in \mathcal{F} : \begin{array}{c} \eta(B \times \Omega) = \mathbb{P}(B) \\ \eta(\Omega \times B) = \mathbb{Q}(B) \end{array} \right\}$$

für $\mathbb{P}, \mathbb{Q} \in \mathcal{P}(\Omega)$. Das definierende Optimierungsproblem wird mit *Monge-Kantorovich Massentransportationsproblem* bezeichnet. Es ist bekannt, daß ζ_c und $\hat{\mu}_c$ für $\mathbb{P}, \mathbb{Q} \in \mathcal{P}_c(\Omega)$ übereinstimmen,

$$\mathcal{P}_c(\Omega) = \left\{ \mathbb{Q} \in \mathcal{P}(\Omega) \middle| \int_\Omega \lambda(\omega)\, \mathbb{Q}[d\omega] < \infty \right\}.$$

Wir beobachten, daß $\mathcal{P}_c(\Omega) \subseteq \mathcal{P}_f(\Omega)$ da

$$|f(\omega, x)| \leq g(\|x\|)\, c(\omega, \omega_0) + |f(\omega_0, x)| \leq g(\rho)(\lambda(\omega) + \lambda(\omega_0)) + |f(\omega_0, x)|$$

für ein fixiertes $\omega_0 \in \Omega$ und $\|x\| \leq \rho$.

Bemerkung. Wenn c die Form

$$c_h(\omega, \tilde{\omega}) := \|\omega - \tilde{\omega}\| \max\{1, h(\|\omega - \omega_0\|), h(\|\tilde{\omega} - \omega_0\|)\}, \qquad \omega, \tilde{\omega} \in \Omega,$$

hat, wobei $\omega_0 \in \Omega$ fixiert ist und $h : \mathbb{R}^+ \to \mathbb{R}^+$ stetig und nichtfallend, so heißt ζ_{c_h} *Fortet-Mourier (artige) Metrik*. Beachte, daß hier nur $\zeta_{c_h} \leq \hat{\mu}_{c_h}$ gilt, da c_h im allgemeinen keine Metrik ist. Im Falle $c(\omega, \tilde{\omega}) = \|\omega - \tilde{\omega}\|$ heißt ζ_c auch *Kantorovich* oder *L^1-Wasserstein Metrik*.

Um den Satz 3.2 mit ζ_c zu verbinden, beachte man, daß $(g(\rho))^{-1} f(\cdot, x) \in F_c$ für alle $x \in \mathbb{R}^r$ mit $\|x\| \leq \rho$ und somit

$$d_{f,\rho}(\mathbb{P}, \mathbb{Q}) \leq g(\rho)\, \zeta_c(\mathbb{P}, \mathbb{Q}) \tag{3.6}$$

[1] Im allgemeinen impliziert $\zeta_c(\mathbb{P}, \mathbb{Q}) = 0$ nicht $\mathbb{P} = \mathbb{Q}$, wenn F_c nicht reichhaltig genug ist.

für $\mathbb{P}, \mathbb{Q} \in \mathcal{P}_f(\Omega)$, da

$$\begin{aligned}
d_{f,\rho}(\mathbb{P}, \mathbb{Q}) &= \sup_{\|x\|\leq\rho} \left| \int_\Omega f(\omega, x)\, \mathbb{P}[d\omega] - \int_\Omega f(\omega, x)\, \mathbb{Q}[d\omega] \right| \\
&\leq \sup_{\|x\|\leq\rho} \sup_{f \in F_{g(\|x\|)c}} \left| \int_\Omega f(\omega)\, \mathbb{P}[d\omega] - \int_\Omega f(\omega)\, \mathbb{Q}[d\omega] \right| \\
&= \sup_{\|x\|\leq\rho} \sup_{f \in F_c} \left| \int_\Omega g(\|x\|) f(\omega)\, \mathbb{P}[d\omega] - \int_\Omega g(\|x\|) f(\omega)\, \mathbb{Q}[d\omega] \right| \\
&\leq g(\rho) \sup_{f \in F_c} \left| \int_\Omega f(\omega)\, \mathbb{P}[d\omega] - \int_\Omega f(\omega)\, \mathbb{Q}[d\omega] \right| \\
&= g(\rho)\, \zeta_c(\mathbb{P}, \mathbb{Q}).
\end{aligned}$$

3.1.3 Szenarioreduktion

Für zweistufige stochastische Programme (3.4), sei angenommen, daß f die Bedingung (3.5) für eine Metrik c und eine nichtfallende Funktion g erfüllt und daß die Wahrscheinlichkeitsverteilung $\mathbb{P}$ zu $\mathcal{P}_c(\Omega)$ gehört. Sei $S(\mathbb{P})$ nichtleer und beschränkt. Dann können wir aus Satz 3.2 Lipschitz-Stabilitätseigenschaften für den Optimalwert $v(\cdot)$ und die ε-approximative Lösungsmenge $S_\varepsilon(\cdot)$ um $\mathbb{P}$ bzgl. der Wahrscheinlichkeits-(semi-)metrik ζ_c ableiten. Wir können also ζ_c als Basis für die Approximation von $\mathbb{P}$ nutzen, d.h. das Prinzip der *optimalen Szenariogenerierung* für (3.4) kann folgendermaßen formuliert werden: Bestimme eine diskrete Menge $\Omega_I = \{\omega_i\}_{i\in I} \subset \Omega$ von Szenarien in Ω von vorgegebener Größe $|I|$ und weise diesen Wahrscheinlichkeiten $\{q_i\}_{i\in I}$ zu, $\sum_{i\in I} q_i = 1$, $q_i \geq 0$, $i \in I$, so daß der Abstand zum Originalmass $\mathbb{P}$ im Sinne der Metrik ζ_c minimal ist,

$$\zeta_c\left(\mathbb{P}, \sum_{i\in I} q_i \delta_{\omega_i}\right) \quad \to \quad \min. \tag{3.7}$$

Für eine gegebene diskrete Approximation $\mathbb{P} = \sum_{i\in I} p_i \delta_{\omega_i}$, $|I| < \infty$, kann man das Prinzip der *optimalen Szenarioreduktion* folgendermaßen formulieren: Bestimme eine Menge $J \subseteq I$ von vorgegebener Größe $|J| = k$ und weise den verbleibenden Szenarien $\Omega_{I\setminus J}$ Wahrscheinlichkeiten $\{q_j\}_{j\notin J}$ zu, $\sum_{j\notin J} q_j = 1$, $q_j \geq 0$, $j \notin J$, so daß

$$\zeta_c\left(\mathbb{P}, \sum_{j\in I\setminus J} q_j \delta_{\omega_j}\right) \quad \to \quad \min. \tag{3.8}$$

Während die Lösung des Problems (3.7) im allgemeinen Fall oft hoffnungslos ist, stellt das Problem (3.8) ein (sehr großes) lineares Problem dar, da $\zeta_c = \hat{\mu}_c$ über $\mathcal{P}_c(\Omega)$ gilt. Wir definieren das Funktional

$$\mathcal{D}(J; q) := \hat{\mu}_c\left(\sum_{i\in I} p_i \delta_{\omega_i}, \sum_{j\in I\setminus J} q_j \delta_{\omega_j}\right)$$

so daß das Problem (3.8) die Form

$$\min\left\{ \mathcal{D}(J;q) \,\middle|\, J \subseteq I, |J| = k, \sum_{j\notin J} q_j = 1, q_j \geq 0 \forall j \right\}$$

bekommt. Für eine gegebene Menge $J \subseteq I$ kann gezeigt werden, daß

$$\mathcal{D}(J) := \min\left\{ \mathcal{D}(J;q) \,\middle|\, \sum_{j\notin J} q_j = 1, q_j \geq 0 \forall j \right\} = \sum_{i\in J} p_i \min_{j\notin J} c(\omega_i, \omega_j) \tag{3.9}$$

gilt und das das Minimum angenommen wird für

$$\bar{q}_j = p_j + \sum_{i \in J : j(i) = j} p_i, \; j \notin J, \quad \text{wobei } j(i) \in \operatorname*{Argmin}_{j \notin J} c(\omega_i, \omega_j), \; i \in J, \tag{3.10}$$

d.h. die Wahrscheinlichkeit p_i eines gelöschten Szenarios ω_i, $i \in J$, wird einem Szenario ω_j, $j \notin J$, zugewiesen, welches sich am nähesten zu ω_i befindet. (*optimale Neuverteilungsregel*)

Es verbleibt, die Menge $J \subseteq I$ mit $|J| = k$ zu finden, die $\mathcal{D}(J)$ minimiert. Da $\min_{J \subseteq I, |J| = k} \mathcal{D}(J)$ ein SetCover-Problem darstellt, welches NP-hart ist, sind effiziente Lösungsalgorithmen kaum zu erwarten. Aber es sind untere und obere Schranken für ein optimales $\mathcal{D}(J)$ bekannt, welche zu folgenden Szenarioreduktionsalgorithmen führen:

Theorem 3.3. *Seien Indizes l_i und u_j rekursiv so gewählt, daß*

$$l_i = \operatorname*{argmin}_{l \in I \setminus \{l_1, \ldots, l_{i-1}\}} p_l \min_{j \neq l} c(\omega_l, \omega_j), \qquad i = 1, \ldots, k, \tag{3.11}$$

$$u_j = \operatorname*{argmin}_{u \in I \setminus \{u_1, \ldots, u_{j-1}\}} \sum_{i \in I \setminus \{u_1, \ldots, u_{j-1}, u\}} p_i \min_{l \in \{u_1, \ldots, u_{j-1}, u\}} c(\omega_l, \omega_i), \qquad j = 1, \ldots, |I| - k. \tag{3.12}$$

Dann gilt

$$\sum_{i=1}^{k} p_{l_i} \min_{j \neq l_i} c(\omega_{l_i}, \omega_j) \leq \min_{J \subseteq I, |J| = k} \mathcal{D}(J) \leq \sum_{i \in I \setminus \left\{u_1, \ldots, u_{|I|-k}\right\}} p_i \min_{j \in \left\{u_1, \ldots, u_{|I|-k}\right\}} c(\omega_i, \omega_j). \tag{3.13}$$

Dieser Satz schlägt zwei Heuristiken für die optimale Reduktion von $|I|$ Szenarien auf $|I| - k$ Szenarien vor. Während der erste eine Menge von k Szenarien bestimmt, die entfernt werden sollen, bestimmt der zweite die $|I| - k$ Szenarien, die behalten werden sollen. (3.11) wird mit *backward reduction* bezeichnet, während (3.12) *forward selection* heißt. Man sieht leicht, daß der backward-reduction Algorithmus ein optimales J im Fall $k = 1$ bestimmt, während der forward selection Algorithmus ein optimales J im Fall $k = |I| - 1$ bestimmt. Algorithmus (3.11) wurde zum sogenannten *simultaneous backward reduction* Algorithmus verbessert ([12]):

$$l_i = \operatorname*{argmin}_{l \in I \setminus \{l_1, \ldots, l_{j-1}\}} \sum_{k \in \{l_1, \ldots, l_{j-1}, l\}} p_k \min_{j \in I \setminus \{l_1, \ldots, l_{j-1}, l\}} c(\omega_k, \omega_l), \qquad i = 1, \ldots, k.$$

Wir haben nun einen Punkt erreicht, an dem wir eine gegebene diskrete Approximation $\mathbb{P}$ um k Szenarien zu einer gröberen Approximation $\mathbb{Q}$ reduzieren können und dabei die verursachte Störung im Optimalwert des konvexen zwei-stufigen stochastischen Programmes mittels

$$|v(\mathbb{P}) - v(\mathbb{Q})| \leq g(\rho) \min_{J \subseteq I, |J| = k} \mathcal{D}(J)$$

abschätzen können, wobei das optimale $\mathcal{D}(J)$ mittels (3.13) beschränkt ist. Eine offene Frage ist die optimale Wahl des Parameters k: Wie viele Szenarien werden benötigt, damit der Abstand $|v(\mathbb{P}) - v(\mathbb{Q})|$ klein ist? Eine Möglichkeit ist es, mit einem kleinen k zu starten und dieses so lange zu erhöhen, wie die untere Schranke für das optimale $\mathcal{D}(J)$, gegeben durch den (simultaneous) backward reduction Algorithmus, "klein genug" ist, oder, auf ähnliche Weise, man startet mit einem großen k und erhöht dessen Wert so lange wie die obere Schranke an das optimale $\mathcal{D}(J)$, gegeben durch den forward selection Algorithmus "zu groß" ist. Aber da wir den Parameter $g(\rho)$ nicht kennen, ist es schwierig, ein quantitatives Maß für "klein genug" oder "zu groß" zu geben. Beachte, daß die Funktion $g(\cdot)$ primär von den Daten des stochastischen Programmes und nur sekundär vom Wahrscheinlichkeitsmaß $\mathbb{P}$ abhängt, wobei $\mathcal{D}(J)$ primär von $\mathbb{P}$ abhängt und die Funktion f nur bei der Definition der Metrik c eine Rolle spielt.

Dies motiviert es, einen Algorithmus zur Szenarioreduktion zu formulieren, welcher ein zweites Maß an die Lücke $|v(\mathbb{P}) - v(\mathbb{Q})|$ ausnutzt. Daher fahren wir damit fort, den abstrakten Algorithmus aus Abschnitt 2.5 für mehrstufige stochastische Programme zu formulieren. Startend mit einer groben Approximation des anfänglichen Szenariobaumes (d.h. wir starten mit einem großen k), verfeinern wir den Szenariobaum (Verkleinerung von k) durch Benutzung des forward-selection Algorithmusses so lange, wie die erreichbare Verbesserung im Optimalwert groß genug zu sein scheint. Um die erreichbare Verbesserung zu berechnen, benutzen wir eine Testverfeinerung der aktuellen Aggregierung. Um sicherzustellen, daß diese Test-Verfeinerung sich nah genug am anfänglichen Baum befindet, schätzen wir den Abstand der Wahrscheinlichkeitsmaße, die durch die Test-Verfeinerung und den anfänglichen Baum induziert werden, mittels (3.13) ab. Aber im Gegenteil zur Szenarioreduktion mittels backward-reduction oder forward-selection, fügen wir Szenarien nur dann zur finalen Approximation hinzu, wenn eine erhebliche Verbesserung der Approximation des stochastischen Programmes erreichbar scheint. Dadurch führt eine zu große Testverfeinerung, verursacht durch eine unangemessene Wahl der oberen Schranke an $g(\rho)\min_J \mathcal{D}(J)$, nicht unmittelbar zu einer Vergrößerung der finalen Approximation.

Da der adaptive Szenarioverfeinerungsalgorithmus mit dem mehrstufigen stochastischen Programm (MSP) in Szenariobaumschreibweise anstatt dessen deterministischen Äquivalent (DA) arbeitet, fahren wir damit fort, (MSP) für diskrete Wahrscheinlichkeitsverteilungen, Szenariobäume und aggregierte Szenariobäume zu formulieren.

3.2 Szenariobaumformulierung

Für das mehrstufige stochastische Programm (MSP), sei angenommen, daß ξ einen endlichen Träger $\Omega_I = \{\omega_i\}_{i\in I} \subseteq \Omega$, $I \subset \mathbb{N}$, besitzt. Dann ist $\mathbb{P} = \sum_{i\in I} p_i \delta_{\omega_i}$ mit $p_i := \mathbb{P}[\{\omega_i\}] > 0$, $\sum_{i\in I} p_i = 1$. Definiere nun die Entscheidungsvariablen $x_{i,t} := x_t(\omega_i)$ und die Realisierungen $\xi_{i,t} := \xi_t(\omega_i)$, $\zeta_{i,t} := \zeta_t(\omega_i)$, $i \in I$. Sei $\mathcal{C}_t$ ein endlicher Erzeuger von $\mathcal{F}_t$, d.h. $\mathcal{F}_t = \sigma(\mathcal{C}_t)$, $\mathcal{C}_t = \{E_{t,j}\}_{j\in J_t}$, so daß $\bigcup_{j\in J_t} E_{t,j} = \Omega_I$, $E_{t,j} \cap E_{t,k} = \emptyset$, $j \neq k$, $t = 1,\ldots,T$. Die Nichtantizipativitätsbedingung $x_t = \mathbb{E}[x_t\,|\mathcal{F}_t]$ liest sich jetzt

$$x_{i,t} = \sum_{j\in J_t}\left(\chi_{E_{t,j}}(\omega_i)\cdot\frac{1}{\mathbb{P}(E_{t,j})}\sum_{\omega_{i'}\in E_{t,j}} p_{i'}x_{i',t}\right) = \sum_{j\in J_t}\left(\chi_{E_{t,j}}(\omega_i)\cdot\sum_{\omega_{i'}\in E_{t,j}} \bar{p}_{i'}x_{i',t}\right) \qquad \forall i \in I$$

mit $\bar{p}_i = \frac{1}{\mathbb{P}(E_{t,j})}p_i$, wobei j so gewählt ist, daß $\omega_i \in E_{t,j}$.

Die Diskretisierung von (MSP) bzgl. dem Wahrscheinlichkeitsmass $\mathbb{P}$ ist definiert als

$$\begin{array}{lllll}
\min\limits_x & \sum\limits_{i\in I} p_i f(x_{i,1},\ldots,x_{i,T},\xi_{i,1},\ldots,\xi_{i,T}) & & & \\
& g_t(x_{i,1},\ldots,x_{i,t},\zeta_{i,t}) \le 0 & & i\in I & t=1,\ldots,T \\
\text{s.d.} & x_{i,t} \in X_t(\zeta_{i,t}) & & i\in I & t=1,\ldots,T \\
& \sum\limits_{\omega_{i'}\in E_{t,j}} \bar{p}_{i'}x_{i',t} = x_{i,t} & j \text{ s.d. } \omega_i \in E_{t,j} & i\in I & t=1,\ldots,T
\end{array} \qquad (\mathrm{MSP}[I])$$

wobei $x = \{x_{i,t}\}_{i\in I, t=1,\ldots,T}$.

Für die Nichtantizipativitätsbedingungen beobachten wir $x_{i,t} = x_{i',t}$ $\forall \omega_i, \omega_{i'} \in E_{t,j}$, $j \in J_t$, $t = 1,\ldots,T$. Explizite Berücksichtigung dieser Restriktionen erlaubt eine Reduktion der Dimension von (MSP$[I]$): Wir konstruieren einen Graphen, dessen Knoten die Mengen $E_{t,j}$, $j \in J_t$, $t = 1,\ldots,T$, sind und mit einer Kante zwischen $E_{t,j}$ und $E_{t+1,k}$ genau dann, wenn $E_{t+1,k} \subseteq E_{t,j}$. Dieser Graph hat die Struktur eines Baumes mit Wurzelknoten $E_{11} = \Omega$ und Blättern $E_{T,i} = \{\omega_i\}$ (da $\mathcal{F}_T = \mathcal{F}$). Sei nun $\bar{N} \subset \mathbb{N}$ eine Nummerierung der Knoten dieses Graphs, so daß der Knoten

$n = 1$ der Wurzelknoten ist. Zur Vereinfachung verzichten wir auf eine extra Notation für die Kanten des Baumes und schreiben einfach $\bar{N}$ für den *anfänglichen Szenariobaum*. Später betrachten wir Teilbäume des anfänglichen Szenariobaumes, aber da deren Struktur durch den Originalbaum induziert wird, sollte das Weglassen einer Beschreibung der Kanten keine Unklarheiten verursachen.

Notation. Für $n \in \bar{N}$ und $N \subseteq \bar{N}$ sei

- n_- der Vorgänger und $N_+(n)$ die Menge der Nachfolger von n
- $n_{-k} := \left(n_{-(k-1)}\right)_-$ mit $n_{-1} := n_-$
- $N_{+k}(n) := N_{+(k-1)}(N_+(n))$ mit $N_{+1}(\cdot) := N_+(\cdot)$
- $t(n)$ so, daß $n_{-t(n)+1} = 1$ (die Tiefe von n)
- $N_t := \{n \in N \,|t(n) = t\}$ alle Knoten mit Tiefe (oder Zeitstufe) t in N
- $p(n) := (1, \ldots, n_-, n)$ der Pfad vom Wurzelknoten zum Knoten n
- $i(n) \in I$ der Szenarioindex, der zu $n \in \bar{N}_T$ gehört.
- $p(i)$ der Pfad, der zum Szenario $i \in I$ gehört: $p(i) := p(n)$ mit $n \in \bar{N}_T$ so, daß $i(n) = i$

Wir definieren rekursiv Knotenwahrscheinlichkeiten $\pi_n := p_{i(n)} = \mathbb{P}\left[\left\{\omega_{i(n)}\right\}\right]$, $n \in \bar{N}_T$, und $\pi_n := \sum_{m \in \bar{N}_+(n)} \pi_m$, $n \in \bar{N} \setminus \bar{N}_T$, so daß $\sum_{n \in \bar{N}_t} \pi_n = 1$, $t = 1, \ldots, T$.

Sei weiterhin $x^n := x_{i(n),t(n)}$, $\xi^n := \xi_{i(n),t(n)} := \xi_{t(n)}\left(\omega_{i(n)}\right)$, $\zeta^n := \zeta_{i(n),t(n)} = \left(\xi^1, \ldots, \xi^{n_-}, \xi^n\right)$. Dann liest sich die Szenariobaumform von (MSP$[I]$) als

$$
\begin{aligned}
\min_x \quad & \sum_{n \in \bar{N}_T} \pi_n f\left(x^{p(n)}, \zeta^n\right) && \\
\text{s.d.} \quad & g_{t(n)}\left(x^{p(n)}, \zeta^n\right) \leq 0 && n \in \bar{N} \\
& x^n \in X_{t(n)}\left(\zeta^n\right) && n \in \bar{N}
\end{aligned}
\tag{MSP$[\bar{N}]$}
$$

wobei $x = \{x^n\}_{n \in \bar{N}}$.

3.3 Aggregierungen und ihre Verfeinerungen

Für mehrstufige stochastische Programme müssen wir die spezielle Struktur der Knotenmenge $\bar{N}$ berücksichtigen, wenn wir eine Partitionierung $\mathcal{N}$ definieren. Partitionierungen werden hier als Aggregierungen von Szenarien betrachten: Wir wählen einige Szenarien $J \subseteq I$ aus, betrachten den durch J induzierten Baum, d.h. die Pfade $\{p(i)\}_{i \in J}$, und definieren eine Zuweisung der verbleibenden Szenarien $I \setminus J$ zu den ausgewählten Szenarien J. Die Partitionierung $\mathcal{N}$ wird nun definiert als die Partitionierung von $\bar{N}_T$, die diese Zuweisung repräsentiert.

Etwas präziser, wir definieren eine Abbildung

$$\text{agg} : \bar{N} \to \bigcup_{i \in J} p(i),$$

so daß $\text{agg}\left(\bar{N}_t\right) \subseteq \bar{N}_t$, $t = 1, \ldots, T$, $\text{agg}\,|_{p(i)} \equiv \text{id}_{p(i)}$, $i \in J$, und die Vorgängeroperation mit agg kommutiert:

$$\text{agg}(n)_- = \text{agg}(n_-) \quad \forall n \in \bar{N} \setminus \{1\}. \tag{3.14}$$

Die Partitionierung $\mathcal{N}$ ist nun definiert als die Menge der Urbilder von agg bzgl. $\bar{N}_T$, d.h.

$$\mathcal{N} = \left\{N \subseteq \bar{N}_T \,|N \in \mathcal{N}\right\} := \left\{\mathrm{agg}^{-1}(n) \,\middle|n \in \bar{N}_T, i(n) \in J\right\}.$$

Die Knoten, die invariant unter einer Anwendung von agg sind, werden als *repräsentativ* bezeichnet, weil sie die ausgewählten Szenarien J im anfänglichen Szenariobaum $\bar{N}$ "repräsentieren". Daher definieren wir $n_T(N)$ für $N \in \mathcal{N}$ als den Knoten $n \in N$ mit $\mathrm{agg}(n) = n$, und weiterhin $n_t(N) := n_T(N)_{-T+t}$, $t = 1, \ldots, T-1$ (Beachte, daß nun auch $\mathrm{agg}(n_t(N)) = n_t(N)$, $t = 1, \ldots, T$.). Sei $p(N) := p(n_T(N)) = (n_1(N), \ldots, n_T(N))$, $N \in \mathcal{N}$. Die Menge aller repräsentativen Knoten wird mit $\bar{N}|_{\mathcal{N}}$ bezeichnet, $\bar{N}|_{\mathcal{N}} = \bigcup_{i \in J} p(i)$. Der Baum, der von $\bar{N}|_{\mathcal{N}}$ induziert wird, heißt *aggregierter Szenariobaum* und wird auch mit $\bar{N}|_{\mathcal{N}}$ bezeichnet. Die Knotenwahrscheinlichkeiten für die Blätter von $\bar{N}|_{\mathcal{N}}$ sind definiert als $\pi_N := \sum_{n \in N} \pi_n$, $N \in \mathcal{N}$.

Wir definieren das aggregierte Problem von (MSP$[\bar{N}]$) durch Beschränkung des Entscheidungsprozesses und der Restriktionen auf die Szenarien J bzw. den aggregierten Szenariobaum. Für die Zielfunktion mitteln wir über die aggregierten Szenarien, was den Vergleich mit der Zielfunktion bzgl. einer Verfeinerung von $\mathcal{N}$ vereinfacht. Das aggregierte Problem liest sich daher

$$\begin{array}{lll}
\min\limits_{x} & \sum\limits_{N \in \mathcal{N}} \sum\limits_{n \in N} \pi_n f\left(x^{p(N)}, \zeta^n\right) & \\
\text{s.d.} & g_{t(n)}\left(x^{p(n)}, \zeta^n\right) \leq 0 & n \in \bar{N}|_{\mathcal{N}} \\
& x^n \in X_{t(n)}(\zeta^n) & n \in \bar{N}|_{\mathcal{N}}
\end{array} \qquad (\text{MSP}[\mathcal{N}])$$

Für die konvexe Relaxierung nehmen wir an, daß $\check{X}_{t(n)}(\zeta^n)$ eine konvexe Obermenge von $X_{t(n)}(\zeta^n)$ ist für alle $n \in \bar{N}$, $\check{f}(\cdot, \zeta^n)$ eine konvexe Unterschätzerfunktion von $f(\cdot, \zeta^n)$ über $\times_{t=1}^{T} \check{X}_t(\zeta^{n_{-T+t}})$ ist für alle $n \in \bar{N}_T$ und $\check{g}_{t(n)}(\cdot, \zeta^n)$ ein konvexe Unterschätzerfunktion von $g_{t(n)}(\cdot, \zeta^n)$ über $\check{X}_{t(n)}(\cdot, \zeta^n)$ ist für alle $n \in \bar{N}|_{\mathcal{N}}$. Dann erhalten wir

$$\begin{array}{lll}
\min\limits_{x} & \sum\limits_{N \in \mathcal{N}} \sum\limits_{n \in N} \pi_n \check{f}\left(x^{p(N)}, \zeta^n\right) & \\
\text{s.d.} & \check{g}_{t(n)}\left(x^{p(n)}, \zeta^n\right) \leq 0 & n \in \bar{N}|_{\mathcal{N}} \\
& x^n \in \check{X}_{t(n)}(\zeta^n) & n \in \bar{N}|_{\mathcal{N}} \\
& c(x) \leq 0 & c \in \mathcal{C}_{\mathcal{N}}
\end{array} \qquad (\text{CMSP}[\mathcal{N}])$$

Für eine Verfeinerung $J' \subseteq I$, $J \subseteq J'$, definieren wir die Abbildung $\mathrm{agg}' : \bar{N} \to \bigcup_{i \in J'} p(i)$ und die Partitionierung $\mathcal{N}'$ von $\bar{N}_T$ so, daß $\bigcup_{i \in J'} p(i) = \bar{N}|_{\mathcal{N}'}$ und $\mathcal{N}' \preceq \mathcal{N}$. Um $\mathcal{N}' \preceq \mathcal{N}$ zu erhalten, fordern wir

$$\mathrm{agg}(\mathrm{agg}'(n)) = \mathrm{agg}(n) \qquad \forall n \in \bar{N}. \tag{3.15}$$

Sei weiterhin $n'_t(N) := \{n_t(N') \,|N' \in \mathcal{N}'|_N\}$, $N \in \mathcal{N}$, die Menge der repräsentativen Knoten in der Verfeinerung von N (zur Zeitstufe t), $t = 1, \ldots, T$. ($n'_t(N)$ ist die Menge der unter agg' invarianten Knoten, welche unter agg auf $n_t(N)$ abgebildet werden.) Die Bedingung $J \subseteq J'$ liefert auch $\bar{N}|_{\mathcal{N}} \subseteq \bar{N}|_{\mathcal{N}'}$.

Wenn das Problem (Proj$[\mathcal{N}, \mathcal{N}']$) für MSPs formuliert wird, so schränken wir einen Entscheidungsprozess auf $\bar{N}|_{\mathcal{N}'}$ auf einen Entscheidungsprozess auf $\bar{N}|_{\mathcal{N}}$ ein, d.h. die $W_N x_N = 0$ Restriktionen haben die Form $x^{n_t(N)} = x^{n_t(N')}$ für alle $N' \in \mathcal{N}'|_N$, $t = 1, \ldots, T$. Eine zusätzliche Bündelung eines Szenarios $i(n_T(N'))$ mit einem Szenario $i(n_T(M))$ für ein $N' \in \mathcal{N}'|_N$ und verschiedenen $N, M \in \mathcal{N}$ wird dabei durch die Bedingungen (3.14) und (3.15) vermieden:

Proposition 3.4. *Seien $N, M \in \mathcal{N}$ und $t \in \{1, \ldots, T\}$. Dann gilt für alle $N' \in \mathcal{N}'|_N$ mit $n_t(N') = n_t(M)$, daß $n_t(N) = n_t(M)$.*

Beweis. Seien $N, M \in \mathcal{N}$, $N' \in \mathcal{N}'|_N$ und $t \in \{1, \ldots, T\}$ so gewählt, daß $n_t(N') = n_t(M)$. Sei $\tau := T - t$. Dann ist

$$\begin{aligned} n_t(M) &= \operatorname{agg}(n_t(M)) \\ &= \operatorname{agg}(n_t(N')) \\ &= \operatorname{agg}(n_T(N'))_{-\tau} \qquad \text{wegen (3.14)} \\ &= (n_T(N))_{-\tau} \qquad \text{wegen } n_T(N') \in N \text{ folgt aus (3.15), daß } \operatorname{agg}(n_T(N')) = n_T(N) \\ &= n_t(N) \end{aligned}$$

□

Wir geben hier noch eine andere Eigenschaft der Abbildung agg an, welche später zu einem Algorithmus führt, der agg basierend auf den Blättern (d.h. Szenarien), ausgewählt aus $\bar{N}_T$, berechnet. Diese Eigenschaft besagt, daß der Abstand eines Knotens n zu $\operatorname{agg}(n)$ nicht länger ist als der Abstand zu anderen repräsentativen Knoten auf der gleichen Zeitstufe. Der Abstand $\operatorname{dist}(n, m)$ für zwei Knoten $n, m \in \bar{N}$ wird hier definiert als die Länge eines kürzesten Pfades zwischen n und m im anfänglichen Baum. Beobachte, daß $\operatorname{dist}(n, m)$ gerade ist für $t(n) = t(m)$.

Proposition 3.5. *Sei die Abbildung* $\operatorname{agg} : \bar{N} \to \bar{N}|_{\mathcal{N}}$ *wie zuvor gegeben. Dann gilt für alle* $n \in \bar{N}$

$$\operatorname{dist}(n, \operatorname{agg}(n)) = \min_{m \in \bar{N}|_{\mathcal{N}} : t(m) = t(n)} \operatorname{dist}(n, m).$$

Beweis. Sei $n \in \bar{N}$ und $\tau \geq 0$, so daß $2\tau = \operatorname{dist}(n, \operatorname{agg}(n))$, d.h. τ ist minimal gewählt mit $n_{-\tau} = \operatorname{agg}(n)_{-\tau}$. Sei weiter $m \in \bar{N}|_{\mathcal{N}}$ mit $t(m) = t(n)$ und $2\tau_* = \operatorname{dist}(n, m)$. Dann haben wir $\tau \leq \tau_*$ zu zeigen.

Wegen $n_{-\tau_*} = \operatorname{agg}(n)_{-\tau_*}$ wissen wir $\operatorname{agg}(n_{-\tau_*}) = n_{-\tau_*}$ (alle Knoten auf einem Pfad von der Wurzel zu einem repräsentativen Knoten sind repräsentativ). Aus (3.14) erhalten wir

$$\operatorname{agg}(n)_{-\tau_*} = \operatorname{agg}(n_{-\tau_*}) = n_{-\tau_*},$$

also ist $\operatorname{dist}(\operatorname{agg}(n), n_{-\tau_*}) = \tau_*$ und damit

$$2\tau = \operatorname{dist}(\operatorname{agg}(n), n) \leq \operatorname{dist}(\operatorname{agg}(n), n_{-\tau_*}) + \operatorname{dist}(n_{-\tau_*}, n) = \tau_* + \tau_*.$$

□

3.3.1 Wahl der Metrik c

Wir identifizieren ein Szenario ω_i, $i \in I$, mit seiner Realisierung $\xi_i = \xi(\omega_i)$ und definieren die Metrik c als eine Fortet-Mourier (artige) Metrik c auf $\times_{t=1}^{T} \mathbb{R}^{s_t}$ mit $\xi(\omega_0) := 0$ und $h \equiv 0$, d.h. $c(\xi_i, \xi_j) := \|\xi_i - \xi_j\|$. Die Norm $\|\cdot\|$ auf $\times_{t=1}^{T} \mathbb{R}^{s_t}$ ist definiert als $\|\xi_i\| := \|\xi_{i,1}\| + \ldots + \|\xi_{i,T}\|$ für gegebene Normen auf $\mathbb{R}^{s_t}$, $t = 1, \ldots, T$.

Weiterhin sei $c_t(\zeta_{i,t}, \zeta_{j,t}) := \sum_{\tau=1}^{t} \|\xi_{i,\tau} - \xi_{j,\tau}\|$, $t = 1, \ldots, T$, so daß $c \equiv c_T$.

3.3.2 Anfängliche Aggregierung

Eine grobe anfängliche Aggregierung ist definiert durch $\mathcal{N} := \{\bar{N}_T\}$, wobei das repräsentative Szenario mittels forward selection (vgl. Formel (3.12)) ausgewählt wird: wähle $n \in \bar{N}_T$, so daß

$$\sum_{m \in \bar{N}_T \setminus \{n\}} p_m c_T(\xi^m, \xi^n) \quad \to \quad \min$$

und setze $\operatorname{agg}(m) := n_{-T+t(m)}$, $m \in \bar{N}$.

Eine anfängliche Verfeinerung $\mathcal{N}'$ definiert sich durch Verfeinerung von $\mathcal{N}$.

3.3.3 Verfeinerung einer Aggregierung

Sei $(\mathcal{N}, \text{agg})$ eine Partitionierung von $\bar{N}_T$ und eine Aggregierung von $\bar{N}$. Sei $\mathcal{M} \subseteq \mathcal{N}$ eine Menge von Blattmengen, die wir verfeinern wollen. Eine Verfeinerung $\mathcal{N}'$ von $\mathcal{N}$ wird dann konstruiert durch Auswahl einer gegebenen Zahl von Szenarien für jedes $M \in \mathcal{M}$ und durch Aktualisierung der Aggregierung für die verbleibenden Szenarien (so daß man eine Abbildung agg$'$ erhält, die die Bedingungen (3.14) und (3.15) erfüllt). Um in Erinnerung zu behalten, welche repräsentativen Knoten hinzugefügt wurden oder zumindest angefasst wurden während der Verfeinerungsoperation, werden diese mit einem Flag versehen (z.B. flag(m) :=wahr). Um eine möglichst gute Reduktion im Abstand zwischen den Wahrscheinlichkeitsmaßen, die durch die Szenariobäume $\bar{N}\,|_{\mathcal{N}}$ und $\bar{N}$ induziert werden, ausgedrückt durch die Wahrscheinlichkeitsmetrik $\hat{\mu}_c$ aus Abschnitt 3.1, zu erreichen, benutzen wir den fast forward selection Algorithmus aus [12], welcher äquivalent ist zum forward selection Algorithmus (3.12), aber eine bessere Laufzeit bzgl. der Auswahl der Szenarien besitzt. Folgend dem Abschnitt 3.1 messen wir mit $\tilde{\mathcal{D}}(N)$, $N \in \mathcal{N}$, den Abstand zwischen dem Szenariobaum, der durch alle Szenarien in N induziert wird, und dem Szenariobaum, welcher nur auf den Szenarien $i(N')$, $N' \in \mathcal{N}'\,|_N$, basiert:

$$\tilde{\mathcal{D}}(N) := \sum_{n \in N \setminus \{n_T(N') | N' \in \mathcal{N}'|_N\}} p_{i(n)} \min_{N' \in \mathcal{N}'|_N} c\left(\xi_{i(n_T(N'))}, \xi_{i(n)}\right).$$

Weiterhin sei

$$\tilde{\mathcal{D}}(\mathcal{N}') := \sum_{N \in \mathcal{N}} \tilde{\mathcal{D}}(N)$$

ein Mass für den Abstand zwischen den Wahrscheinlichkeitsmaßen, die durch die Bäume $\bar{N}\,|_{\mathcal{N}'}$ und $\bar{N}$ induziert werden. Beachte, daß dieses Mass nicht völlig konsistent mit dem Mass $\mathcal{D}(J)$, definiert in (3.9), ist, da der forward-selection Algorithmus beschränkt wird auf Teilbäume von $\bar{N}$, wenn eine Verfeinerung einer Aggregierung $N \in \mathcal{N}$ bestimmt wird, d.h. die Minimierung in (3.12) wird beschränkt auf Teilmengen von I. Weiterhin folgen wir, der algorithmischen Effizienz wegen, nicht genau der optimalen Neuverteilungsregel (3.10). Aber man kann $\tilde{\mathcal{D}}(\mathcal{N}')$ als eine obere Schranke für

$$\mathcal{D}(I \setminus \{i(N') | N' \in \mathcal{N}'\}) = \sum_{n \in \bar{N}_T \setminus \{n_T(N') | N' \in \mathcal{N}'\}} p_{i(n)} \min_{N' \in \mathcal{N}'} c\left(\xi_{i(n_T(N'))}, \xi_{i(n)}\right)$$

betrachten, da $\bigcup_{N \in \mathcal{N}} (N \setminus \{n_T(N') | N' \in \mathcal{N}'|_N\}) = \bar{N}_T \setminus \{n_T(N') | N' \in \mathcal{N}'\}$.

Algorithmus 3.6 (Szenariobaumverfeinerung).

Input: Ein aggregierter Szenariobaum $(\mathcal{N}, \text{agg})$, eine Menge $\mathcal{M} \subseteq \mathcal{N}$ von Blattmengen zur Verfeinerung und Grenzen $s_M \in \{1, \ldots, |M|\}$, $M \in \mathcal{M}$.

Compute:

agg$'$:= agg, $\mathcal{N}' := \mathcal{N} \setminus \mathcal{M}$
for all $M \in \mathcal{M}$ **do**
  $J := M \setminus \{n_T(M)\}$ {zu löschende Szenarien, aber behalte $i(n_T(M))$ auf jeden Fall}
  Definiere Szenarioabstände $c_{n,k} := \min\left\{c_T\left(\zeta^n, \zeta^{n_T(M)}\right), c_T\left(\zeta^n, \zeta^k\right)\right\}$, $n, k \in J$.
  while $|J| \geq |M| - s_M$ **do**
    Wähle $m \in J$, so daß

$$\mathcal{D} := \sum_{n \in J \setminus \{m\}} p_n c_{n,m}$$

    minimiert wird und setze $J := J \setminus \{m\}$, d.h. behalte das Szenario m.
    Aktualisiere die Szenarioabstände: $c_{n,k} := \min\{c_{n,k}, c_{n,m}\}$, $n, k \in J$.
  end while
  for all $m \in M \setminus J$ **do**
    agg$'(m) := m$
    flag(m') :=wahr

    end for
    Aktualisiere agg' durch Aufruf von aggregate$(M \setminus J, (n_T(M)))$.
    $\mathcal{N}' := \mathcal{N}' \cup \left\{\mathrm{agg}'^{-1}(m) \,|\, m \in M \setminus J\right\}$
    $\tilde{\mathcal{D}}(M) := \mathcal{D}$
  end for
Output: $\mathcal{N}'$, agg' , $\tilde{\mathcal{D}}(\mathcal{N}')$

Der Algorithmus aggregate(M^t, p) aktualisiert die Aggregierung eines Teilbaumes von $\bar{N}$, dessen Wurzel ein Vorgänger eines Knotens in M^t ist, wobei M^t eine Menge von repräsentativen Knoten zur Zeitstufe t ist. Die Aggregierungen werden so berechnet, daß die resultierende Abbildung agg' die Bedingungen (3.14) und (3.15) erfüllt. Die Aggregierung eines Knotens $n \in M^t$ wird aktualisiert sobald entweder n_- repräsentativ gemacht wird (Algorithmus 3.7, Zeile 8) oder seine Aggregierung aktualisiert wird (Algorithmus 3.8, Zeile 4), aber nur in dem Fall, daß $\mathrm{agg}(n) \in p(M)$ gilt, d.h. wir aktualisieren nur die Aggregierung von Knoten welche zu dem Pfad aggregiert waren, den wir im Moment verfeinern. Dies, und die Bedingung, daß ein Pfad nur mit Pfaden gebündelt werden kann, die zu Blättern in M führen (dafür steht das Flag), stellt die Bedingung (3.15) sicher. Die Idee, eine Aggregierung von n zu aktualisieren, sobald die Aggregierung des Vorgängers aktualisiert wird, entspringt der kürzeste-Pfad-Eigenschaft, die in Proposition 3.5 gezeigt wurde.

Algorithmus 3.7 (Knotenaggregierung). aggregate(M^t, p):
Input: eine Menge $M^t \subseteq \bar{N}_t$ von repräsentativen Knoten und ein Referenzpfad $p = (p_t, \ldots, p_T)$
Compute:
1: **if** $t = 1$ **then**
2:   **return**
3: **end if**
4: **for all** $m \in M^t$ **do**
5:   Definiere m_- als repräsentativ: $\mathrm{agg}'(m_-) := m_-$
6:   flag(m_-) :=wahr
7:   **for all** $n \in N_+(m_-)$ mit $\mathrm{agg}(n) = p_t$, die nicht repräsentativ sind **do**
8:     aktualisiere $\mathrm{agg}'\big|_{N_*(n)}$ durch Aufruf von bundle$(m_-, n, (p_{t+1}, \ldots, p_T))$ wobei $N_*(n) := \bigcup_{\tau=0}^{T-t(n)} N_{+\tau}(n)$ der Teilbaum am Knoten n in $\bar{N}$ ist.
9:   **end for**
10: **end for**
11: $M^{t-1} := \{m_- \,|\, m \in M^t\}$
12: fahre fort mit der Aktualisierung von agg' durch Aufruf von aggregate$\big(M^{t-1}, \big((p_t)_-, p_t, \ldots, p_T\big)\big)$
Output: eine aktualisierte Aggregierung

Algorithmus 3.8 (Knotenbündelung). bundle(m_-, n, p):
Input: ein repräsentativer Knoten m_-, ein Knoten n, dessen Aggregierung aktualisiert werden soll, und ein Referenzpfad $\big(p_{t(n)+1}, \ldots, p_T\big)$
Compute:
1: wähle einen repräsentativen Nachfolger $m \in N_+(m_-)$ mit flag(m) =wahr, so daß $c_{t(n)}(\zeta^n, \zeta^m)$ minimiert wird
2: bündele n mit m: $\mathrm{agg}'(n) := m$.
3: **for all** $n_+ \in N_+(n)$ mit $\mathrm{agg}(n_+) = p_{t(n)+1}$ **do**
4:   bundle$\big(m, n_+, \big(p_{t(n)+2}, \ldots, p_T\big)\big)$
5: **end for**
Output: eine aktualisierte Aggregierung für $N_*(n)$

Proposition 3.9. *Die durch Algorithmus 3.6 berechnete Abbildung* agg' *erfüllt die Bedingungen (3.14) und (3.15).*

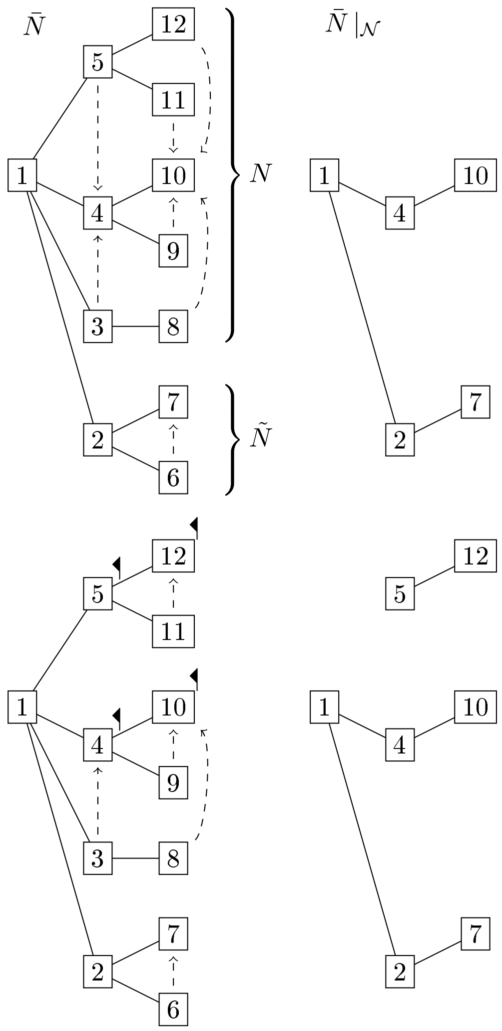
Beweis. agg' wird modifiziert in Zeile 5 von Algorithmus 3.7 und Zeile 2 von Algorithmus 3.8.

Zur Bedingung (3.14): Zeile 5 von Algorithmus 3.7 stellt die Invarianz der Knoten $\bar{N}|_{\mathcal{N}'}$ bzgl. agg' sicher. In Zeile 2 von Algorithmus 3.8 wird n mit einem Knoten $m \in N_+(m_-)$ mit gesetztem $\mathrm{flag}(m)$ aggregiert. n_- wurde entweder in Zeile 2 von Algorithmus 3.8 durch einen vorangegangenen Aufruf oder durch Zeile 5 in Algorithmus 3.7 aggregiert. In beiden Fällen wurde n_- mit m_- aggregiert, d.h. $\mathrm{agg}'(n_-) = m_-$. Verbindet man dies mit $\mathrm{agg}'(n) \in N_+(m_-)$, so erhält man $\mathrm{agg}'(n)_- = m = \mathrm{agg}'(n_-)$.

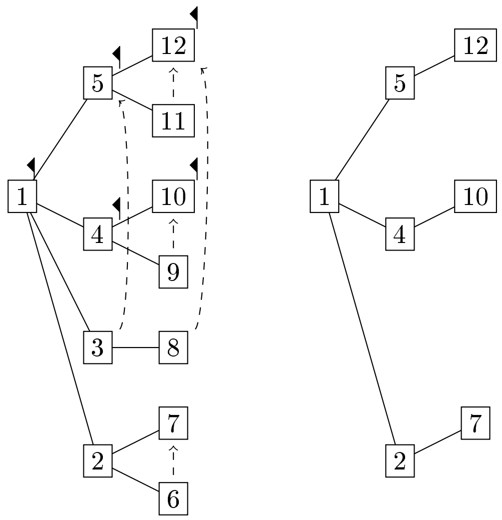
Zur Bedingung (3.15): Zeile 7 von Algorithmus 3.7 und Zeile 3 von Algorithmus 3.8 stellen sicher, daß nur Aggregierungen von Knoten n mit $\mathrm{agg}(n) = n_{t(n)}(M)$ aktualisiert werden (es gilt $p_{t(n)} = n_{t(n)}(M)$, da $p_T := n_T(M)$ aus Algorithmus 3.6). Aufgrund der Eigenschaft, daß ein Knoten n nur mit einem repräsentativen Knoten gebündelt wird, dessen Flag gesetzt ist, ist $\mathrm{agg}(\mathrm{agg}'(n)) = n_{t(n)}(M)$ sichergestellt. □

Wir beenden diesen Abschnitt mit einem erläuternden Beispiel des Verfeinerungsalgorithmusses 3.6:

Beispiel. Betrachte einen anfänglichen Szenariobaum $\bar{N}$ auf 12 Knoten und 3 Zeitstufen. Sei $\mathcal{N} := \{N, \tilde{N}\}$ mit $N := \{8, 9, 10, 11, 12\}$, $\tilde{N} := \{6, 7\}$, $n_3(\{6, 7\}) = 7$ und $n_3(\{8, 9, 10, 11, 12\}) = 10$ eine gegebene Aggregierung. Die Abbildung agg ist mit gestrichelten Pfeilen angedeutet.
Wir möchten $\mathcal{M} := \{N\}$ mit $s_N = 1$ verfeinern. Der Algorithmus 3.6 beginnt mit der Auswahl eines Knotens $m \in J := N \setminus \{10\}$, so daß $\sum_{n \in J \setminus \{m\}} p_n c_{n,m}$ minimiert wird. Sei das Minimum angenommen für $m = 12$, also $M \setminus J = \{10, 12\}$. Für die Aggregierungsabbildung setzen wir $\mathrm{agg}'(12) := 12$ und das Flag wird für die Knoten 10 und 12 gesetzt. Dann wird der Algorithmus 3.7 mit $M^3 = \{10, 12\}$ und $p = (p_3) = (10)$ aufgerufen.
Algorithmus 3.7 fügt den Vorgänger von Knoten $m = 12$ (Knoten $m_- = 5$) zum aggregierten Baum durch Setzen von $\mathrm{agg}'(5) := 5$ hinzu und setzt das Flag von Knoten 5. Dann wird für alle Knoten $n_+ \in N_+(5) = \{11, 12\}$, welche nicht repräsentativ sind und die zum Knoten $p_3 = 10$ aggregiert sind, d.h. für $n_+ = 11$, der Algorithmus 3.8 mit $m_- = 5$, $n = 11$ und $p = ()$ aufgerufen.
Hier wird ein repräsentativer Knoten $m \in N_+(5) = \{11, 12\}$ mit gesetztem Flag ausgewählt, der den Abstand $c_3(\zeta^{11}, \zeta^m)$ minimiert. Die einzige Möglichkeit ist $m = 12$, so daß $\mathrm{agg}'(11)$ auf 12 gesetzt wird. $N_+(11)$ ist leer, so daß Algorithmus 3.8 beendet ist und wir kehren zurück zu Zeile 4 in Algorithmus 3.7.
Jetzt werden Zeilen 4 bis 10 für $m = 10$ ausgeführt: Das Flag für Knoten 4 wird gesetzt und die Aggregierung von Knoten 9 überdacht, was aber zu $\mathrm{agg}'(9) = 10$ führt. Als nächstes ruft sich Algorithmus 3.7 selbst mit $M^2 := \{4, 5\}$ und $p = (5, 12)$ auf.



Für $m = 5$ (in Zeile 4) versucht der Algorithmus den Knoten 1 zum aggregierten Baum hinzuzufügen ($\text{agg}'(1) := 1$), und das Flag von Knoten 1 wird gesetzt. Aus Zeile 7-8 sehen wir, daß der Algorithmus 3.8 für $n = 3$ (mit $m_- = 1$ und $p = (12)$) aufgerufen wird.
Hier werden die repräsentativen Knoten $m \in N_+(1) = \{2, 3, 4, 5\}$ mit gesetztem Flag überprüft. Dies sind die Knoten 4 und 5. Angenommen, $c_2(\zeta^3, \zeta^5) < c_2(\zeta^3, \zeta^4)$, so daß $m = 5$ ausgewählt wird. $\text{agg}'(3)$ wird auf 5 gesetzt und die bundle Methode wird für alle $n_+ \in N_+(3) = \{8\}$ aufgerufen, mit $m_- = 5$ und leerem $p = ()$.
Überprüfung der Aggregierung von Knoten 8 resultiert in $\text{agg}'(8) := 12$, der einzigen Möglichkeit in diesem Fall.
Zurück in Algorithmus 3.7, die Zeilen $5 - 9$ für $m = 4$ (wir hatten $M^2 = \{4, 5\}$) ändern den Graph nicht. Letztendlich wird Algorithmus 3.7 mit $M^1 = \{1\}$ und $p = (1, 5, 12)$ aufgerufen. Dies beendet die Verfeinerungsprozedur.



3.4 Dekomposition

Neben den Aggregierungs- und Verfeinerungsalgorithmen besteht eine zweite wichtige Komponente im Szenarioreduktionsalgorithmus aus konvexen Relaxierungen und Schnittgenerierung. In Abschnitt 2.4.2 haben wir ein Spaltengenerierungsverfahren, basierend auf Lagrange-Dekomposition, zur Berechnung von Lagrange-Schnitten vorgeschlagen. Solche dualen Dekompositionsmethoden nutzen die separabele Struktur von Optimierungsproblemen aus. Restriktionen, welche zu dieser separablen Struktur im Widerspruch stehen, werden zur Zielfunktion hinzugefügt, versehen mit dualen Variablen. Für mehrstufige stochastische Programme scheinen drei duale Dekompositionsverfahren vielversprechend zu sein: *Szenario Dekomposition* dualisiert die Nichtantizipativitätsrestriktionen von (MSP), so daß die diskretisierte Form (MSP$[I]$) in viele kleinere Probleme zerfällt, eines für jedes Szenario. Entkopplung der Zeitstufen führt zur sogenannten *nodalen Dekomposition*, welche die zeitkoppelnden Restriktionen $g_t(x_1, \ldots, x_t, \zeta_t)$ von (MSP) dualisiert. In [6] wurde gezeigt, daß die Szenariodekomposition eine kleinere Dualitätslücke bietet als die nodale Dekomposition. Ein dritter Ansatz basiert auf einer separablen Struktur, die nicht durch die Stochastik oder die Zeit-Diskretisierung des Problemes, sondern durch das Optimierungsmodell selbst gegeben ist. Große Modelle mit Netzwerkstruktur bieten oft solche Separabilität (siehe Abschnitt 3.6.3 für ein Beispiel) und manchmal sind spezialisierte Lösungsalgorithmen für die Teilprobleme verfügbar, die es erlauben, den Lösungsprozess zu beschleunigen. Diese so genannte *geographische Dekomposition* wird unser Favorit sein, aber aus Gründen der Vollständigkeit geben wir zunächst eine Formulierung der anderen beiden Methoden an. Vergleiche der Dualitätslücken zwischen geographischer Dekomposition und Szenario- bzw. nodaler Dekomposition findet man in [6].

Neben diesen dualen Dekompositionsverfahren, wurde die sogenannte "nested Benders decomposition" erfolgreich auf lineare mehrstufige stochastische Programme angewendet ([5]). Ähnlich wie die nodale Dekomposition relaxiert sie die zeit-koppelnden Restriktionen von (MSP$[\bar{N}]$) um Teilprobleme zu erhalten, die den Knoten des Szenariobaumes angefügt sind. Wir geben eine kurze Einführung in diese Dekompositionsmethode am Ende dieses Kapitels.

3.4.1 Szenario Dekomposition

Betrachte das mehrstufige stochastische Optimierungsproblem (MSP). Seien $\lambda_t \in L^1(\Omega, \mathcal{F}, \mathbb{P}; \mathbb{R}^{r_t})$, $t = 1, \ldots, T$, L^1-beschränkte $\mathcal{F}$-messbare Funktionen. Dann ist die Szenariodekomposition definiert durch Dualisierung der Nichtantizipativitätsbedingungen,

$$\begin{array}{cll} \max\limits_{\lambda=(\lambda_t)_{t=1}^T} \min\limits_x & L_{\mathrm{szen}}(x,\xi;\lambda) & \\ \text{s.d.} & g_t(x_1,\ldots,x_t,\zeta_t) \leq 0 & \mathbb{P}\text{-f.s.} \quad t=1,\ldots,T \\ & x_t \in X_t(\zeta_t) & \mathbb{P}\text{-f.s.} \quad t=1,\ldots,T \end{array}$$

mit der Lagrangefunktion

$$L_{\mathrm{szen}}(x,\xi;\lambda) := \mathbb{E}\left[f(x,\xi) + \sum_{t=1}^T \langle \lambda_t, x_t - \mathbb{E}[x_t | \mathcal{F}_t]\rangle\right].$$

Wir beobachten

$$\begin{aligned} \mathbb{E}[\langle \lambda_t, \mathbb{E}[x_t|\mathcal{F}_t]\rangle] &= \mathbb{E}[\mathbb{E}[\langle \lambda_t, \mathbb{E}[x_t|\mathcal{F}_t]\rangle |\mathcal{F}_t]] \\ &= \mathbb{E}[\langle \mathbb{E}[\lambda_t|\mathcal{F}_t], \mathbb{E}[x_t|\mathcal{F}_t]\rangle] \\ &= \mathbb{E}[\mathbb{E}[\langle \mathbb{E}[\lambda_t|\mathcal{F}_t], x_t\rangle |\mathcal{F}_t]] \\ &= \mathbb{E}[\langle x_t, \mathbb{E}[\lambda_t|\mathcal{F}_t]\rangle] \end{aligned}$$

für $t = 1, \ldots, T$, so daß die Lagrangefunktion geschrieben werden kann als

$$L_{\mathrm{szen}}(x,\xi;\lambda) := \mathbb{E}\left[f(x,\xi) + \sum_{t=1}^T \langle x_t, \lambda_t - \mathbb{E}[\lambda_t | \mathcal{F}_t]\rangle\right].$$

Angenommen, die Verteilung $\mathbb{P}$ ist diskret, d.h. wir gehen über von (MSP) zu (MSP[I]), basierend auf den Szenarien Ω_I, so ergibt sich das duale Problem

$$\max_\lambda \quad \mathrm{val}(D_\lambda[I]) \tag{3.16}$$

wobei die duale Funktion $\lambda \mapsto \mathrm{val}(D_\lambda[I])$ definiert ist über das Lagrangeproblem

$$\begin{array}{clll} \min\limits_x & L_I(x,\xi;\lambda) & & \\ \text{s.d.} & g_{t,i}(x_{1,i},\ldots,x_{t,i},\zeta_{t,i}) \leq 0 & i \in I, & t=1,\ldots,T \\ & x_{t,i} \in X_t(\zeta_{t,i}) & i \in I, & t=1,\ldots,T \end{array} \tag{$D_\lambda[I]$}$$

mit Lagrangefunktion

$$L_I(x,\xi;\lambda) := \sum_{i\in I} p_i \left(f(x_i,\xi_i) + \sum_{t=1}^T x_{i,t}\left(\lambda_{i,t} - \sum_{\omega_{i'}\in E_{t,j(i,t)}} \bar{p}_{i'}\lambda_{i',t}\right)\right),$$

$\lambda_{i,t} \in \mathbb{R}$ und $j(i,t) := j$, so daß $\omega_i \in E_{t,j}$ (wir erinnern uns an die Nichtantizipativitätsbedingungen für den diskreten Fall aus Abschnitt 3.2). Dann lesen sich die partiellen Lagrangeprobleme als

$$\begin{array}{cll} \min\limits_{x_i} & f(x_i,\xi_i) + \sum\limits_{t=1}^T x_{i,t}\left(\lambda_{i,t} - \sum\limits_{\omega_{i'}\in E_{t,j(i,t)}} \bar{p}_{i'}\lambda_{i',t}\right) & \\ \text{s.d.} & g_t(x_{i,1},\ldots,x_{i,t},\zeta_{i,t}) \leq 0 \quad t=1,\ldots,T & \\ & x_{i,t} \in X_t(\zeta_{i,t}) \quad t=1,\ldots,T & \end{array} \quad i \in I. \tag{$D_{\lambda,i}$}$$

Der stochastische Szenariodekompositionsalgorithmus von Higle, Rayco und Sen

Für eine Erläuterung einer möglichen Anwendung der Szenariodekomposition geben wir hier eine kurze Beschreibung des "Stochastic Scenario Decomposition Algorithm", vorgeschlagen in [16], an. Er basiert auf der Generierung von Schnittebenen, um eine stückweise lineare konkave Approximation der dualen Funktion $\lambda \mapsto \mathrm{val}(D_\lambda[I])$ zu entwickeln. Weiterhin wird die Menge der Szenarien I und Wahrscheinlichkeiten p_i, $i \in I$, ersetzt durch eine empirische Verteilung, welche es erlaubt, mit einer kleinen Anzahl von Szenarien zu starten und während des Approximationsprozesses der dualen Funktion weitere Szenarien hinzuzufügen (oder die Wahrscheinlichkeit von bereits ausgewählten Szenarien zu aktualisieren). Um Unbeschränktheit der Lösungsmenge des dualen Problemes zu vermeiden, wird die Menge der zulässigen dualen Variablen beschränkt auf solche, die die "Null-Erwartungswert-Restriktionen" $\mathbb{E}[\lambda_t | \mathcal{F}_t] \equiv 0$, $t = 1, \ldots, T$, erfüllen, d.h. das duale Problem (3.16) wird ersetzt durch

$$\begin{array}{ll} \max\limits_{\lambda} & \mathrm{val}(D_\lambda[I]) \\ \text{s.d.} & \sum\limits_{i' \in I:\, \omega_{i',t} \in E_{t,j}} \bar{p}_{i'} \lambda'_{i',t} = 0 \quad j \in J_t, \quad t = 1, \ldots, T. \end{array} \tag{3.17}$$

Beachte, daß die zusätzlichen Restriktionen $\sum_{i' \in I:\, \omega_{i',t} \in E_{t,j}} \bar{p}_{i'} \lambda'_{i',t} = 0$, $j \in J_t$, $t = 1, \ldots, T$, den Wert val (3.17) nicht beeinflussen, denn

$$\lambda_{i,t} - \sum_{\omega_{i'} \in E_{t,j(i,t)}} \bar{p}_{i'} \lambda_{i',t} = (\lambda_{i,t} - \kappa) - \sum_{\omega_{i'} \in E_{t,j(i,t)}} \bar{p}_{i'} (\lambda_{i',t} - \kappa)$$

in $L_I(x, \xi; \lambda)$, so daß wenn $\sum_{i' \in I:\, \omega_{i',t} \in E_{t,j}} \bar{p}_{i'} \lambda'_{i',t} = \kappa$ für ein $j \in J_t$ und $\kappa \in \mathbb{R}$, dann erhält man mit $\tilde{\lambda}_{i',t} := \lambda_{i',t} - \kappa$, $i' \in I$ so daß $\omega_{i',t} \in E_{t,j}$, die Gleichheit $\mathrm{val}\left(D_{\tilde{\lambda}}[I]\right) = \mathrm{val}(D_\lambda[I])$ und erfüllt zusätzlich die Restriktion $\sum_{i' \in I:\, \omega_{i',t} \in E_{t,j}} \bar{p}_{i'} \tilde{\lambda}'_{i',t} = 0$.

Wir fahren fort mit der Beschreibung einer Iteration des Algorithmusses. Seien I die aktuelle Menge von Szenarien und p_i, $i \in I$, die entsprechenden Wahrscheinlichkeiten. Die aktuelle Menge $\mathcal{C}$ von Schnitten sei durch Tupel $\left(\alpha_i, (\beta_{i,t})_{t=1}^T\right)_{i \in I} \in \times_{i \in I} (\mathbb{R} \times \mathbb{R}^r)$ beschrieben. Sei λ die letzte duale Schätzung und $\bar{\lambda}$ die vorläufige Lösung.

Algorithmus 3.10 (Stochastische Szenariodekomposition [16]).
Compute:
  Bestimme Lösungen x_i, $i \in I$, der partiellen Lagrangeprobleme $(D_{\lambda,i})$, $i \in I$.
  Aktualisiere das Master-Problem:
    Generiere neue Schnitte mittels $\beta_{i,t} := p_i x_{i,t}^T$, $t = 1, \ldots, T$, und $\alpha_i := p_i f(x_i, \xi_i)$, $i \in I$.
    for all $i \in I$ **do** {Aktualisiere alle zuvor generierten Schnitte}
      if i wurde gerade eben zu I hinzugefügt **then**
        for all $(\hat{\alpha}_j, (\hat{\beta}_{j,t})_{t=1}^T)_{j \in I \setminus \{i\}} \in \mathcal{C}$ **do**
          if $(\hat{\alpha}_j, (\hat{\beta}_{j,t})_{t=1}^T)_{j \in I \setminus \{i\}}$ wurde unter Benutzung der vorläufigen Lösung generiert **then**
            bestimme einen Lösungspunkt $\bar{x}_i$ des zugehörigen Teilproblemes $\left(D_{\bar{\lambda},i}\right)$
            $\hat{\alpha}_i := p_i f(\bar{x}_i, \xi_i)$, $\hat{\beta}_{i,t} := p_i \bar{x}_{i,t}^T$, $t = 1, \ldots, T$
          else
            $\hat{\alpha}_i := \alpha_i$, $\hat{\beta}_{i,t} := \beta_{i,t}$, $t = 1, \ldots, T$
          end if
        end for
      else {i war bereits bekannt}
        aktualisiere alle Schnitte in $\mathcal{C}$ auf die aktuelle Wahrscheinlichkeitsverteilung $(p_i)_{i \in I}$.
      end if
    end for

Teste die vorläufige Lösung:
Sei $\nu(\lambda) := \min_{(\alpha_i,(\beta_{i,t}))\in\mathcal{C}} \sum_{i\in I}\sum_{t=1}^{T} \alpha_{i,t} - \beta_{i,t}\lambda_{i,t}$.
if der neue Schnitt hat den Abstand $\nu(\lambda) - \nu(\bar{\lambda})$ ausreichend verbessert **then**
mache λ zur neuen vorläufigen Lösung, $\bar{\lambda} := \lambda$
end if

Erhalte einen neuen dualen Schätzer durch Lösen des Master-Problems ($\sigma > 0$ ist ein gegebenen Parameter)

$$\begin{array}{ll} \max\limits_{\lambda,\nu} & \nu - \frac{\sigma}{2}\left\|\lambda - \bar{\lambda}\right\| \\ & \sum\limits_{i\in I}\sum\limits_{t=1}^{T} \alpha_{i,t} - \beta_{i,t}\lambda_{i,t} \geq \nu \quad \left(\alpha_i, (\beta_{i,t})_{t=1}^{T}\right)_{i\in I} \in \mathcal{C} \\ \text{s.d.} & \\ & \sum\limits_{i'\in I:\, \omega_{i',t}\in E_{t,j}} \bar{p}_{i'}\lambda_{i',t} = 0 \quad j \in J_t, \quad t = 1,\ldots,T. \end{array}$$

Aktualisierung der empirischen Verteilung:
Generiere ein Szenario zufällig und unabhängig von allen bisher generierten Szenarios und füge es zu I hinzu. Setze den zugehörigen dualen Multiplikator auf 0, falls das Szenario zuvor nicht bekannt war. Aktualisiere p_i, $\bar{p}_i$, $i \in I$.
Projiziere die vorläufige Lösung $\bar{\lambda}$, so daß die Null-Erwartungswert-Bedingungen im Master-Problem erfüllt sind.

Das folgende Resultat zur asymptotischen Konvergenz des stochastischen Szenariodekompositionsalgorithmusses wurde gezeigt ([16, Theorem 6]):

Satz. *Angenommen, das primale Problem* (MSP[I]) *ist zulässig und hat "relatively complete recourse" (d.h. das duale Problem zu* (MSP[I]) *ist beschränkt für alle zulässigen Entscheidungen* $(x_{i,1},\ldots,x_{i,t})$, $t = 1,\ldots,T$*). Wenn* $\operatorname{feas}(D_{\lambda,i})$ *kompakt ist für alle* $i \in I$, I *endlich, und die Folge der dualen Schätzungen* λ *beschränkt ist, dann, mit Wahrscheinlichkeit Eins, existiert ein Häufungspunkt der Folge von vorläufigen Lösungen* $\bar{\lambda}$*, der eine optimale Lösung von* (3.17) *ist.*

Aus computertechnischer Sicht beachte man, daß in jeder Iteration des Algorithmusses eine neue Menge von Variablen λ_i (wenn i ein "neues" Szenario ist) und eine neue Schnittebene in das Master-Problem eingefügt werden. Dies führt zu einem signifikanten Wachstum des Master-Problems während des Iterations-Prozesses. Deshalb wurde in [16] ein Verfahren vorgeschlagen, welches Variablen aggregiert, deren zugehörige Daten im Master-Problem ähnlich sind. Diese Aggregierung variiert von einer Iteration zur nächsten, besonders wenn sich die vorläufige Lösung ändert. Sie wird zunächst bestimmt durch Generierung von Clustering-Vektoren c_j für jede Komponente j von $\lambda - \bar{\lambda}$. Die Komponenten von c_j sind

- der Wert der zugehörigen Komponente in der vorläufigen Lösung, $\bar{\lambda}_j$
- der Koeffizient der Variablen in dem Schnitt, der in der vorläufigen Lösung gebildet wurde
- der Koeffizient der Variablen im zuletzt gebildeten Schnitt
- der aggregierte Koeffizient der Variablen in allen anderen Schnittebenen, die im Master-Problem auftreten

Dann wird die kleinste Box, die alle Vektoren c_j enthält, partitioniert und alle Komponenten von $\lambda - \bar{\lambda}$, deren Clustering-Vektoren in der gleichen Partition enthalten sind, werden innerhalb des Master-Problems aggregiert (durch Mittelung). Eine Abschätzung für den relativen Fehler durch dieses Aggregierungsverfahren ist in [16] angegeben, so daß eine Partitionierung verfeinert werden kann, wenn die Fehlerschätzung eine gegebene Toleranz überschreitet.

3.4.2 Nodale Dekomposition

Wir betrachten das mehrstufige Optimierungsproblem (MSP) und nehmen an, daß die Zielfunktion f und zeit-koppelnden Restriktionen g_t zerlegt werden können in eine Summe von Funktionen, von denen jede nur von einer Zeitstufe abhängt,

$$\begin{aligned} f(x,\xi) &= \sum_{t=1}^{T} f_t(x_t,\xi), \\ g_t(x_1,\ldots,x_t,\zeta_t) &= \sum_{\tau=1}^{t} g_{t,\tau}(x_\tau,\zeta_t), \end{aligned}$$

mit $f_t : \mathbb{R}^{r_t} \times \mathbb{R}^s \to \mathbb{R}$ und $g_{t,\tau} : \mathbb{R}^{r_\tau} \times \mathbb{R}^{s_t} \to \mathbb{R}^{d_t}$, $\tau = 1,\ldots,t$, $t = 1,\ldots,T$. Seien weiterhin $\lambda_t \in L^1\left(\Omega,\mathcal{F},\mathbb{P};\mathbb{R}^{d_t}\right)$, $t = 1,\ldots,T$, L^1-beschränkte, $\mathcal{F}$-messbare Funktionen. Dann ist die nodale Dekomposition von (MSP) definiert als

$$\begin{array}{lll} \displaystyle\max_{\lambda_t:\,\lambda_t\geq 0\ \mathbb{P}\text{-f.s.}}\ \min_{x} & L_{\text{nod}}(x,\xi;\lambda) & \\ \text{s.d.} & x_t \in X_t(\zeta_t) \quad \mathbb{P}\text{-f.s.} & t=1,\ldots,T \\ & x_t = \mathbb{E}\left[x_t \,|\mathcal{F}_t\right] & t=1,\ldots,T \end{array}$$

wobei

$$L_{\text{nod}}(x,\xi;\lambda) := \sum_{t=1}^{T} \mathbb{E}\left[f_t(x_t,\xi) + \sum_{\tau=1}^{t} \langle \lambda_\tau, g_{t,\tau}(x_\tau,\zeta_t)\rangle\right].$$

Damit erhalten wir die partiellen Lagrangeprobleme

$$\left.\begin{array}{ll} \displaystyle\min_{x_t} & \mathbb{E}\left[f_t(x_t,\xi) + \left\langle \lambda_t, \sum_{t'=1}^{T} g_{t',t}(x_t,\zeta_{t'})\right\rangle\right] \\ \text{s.d.} & x_t \in X_t(\zeta_t) \quad \mathbb{P}\text{-f.s.} \\ & x_t = \mathbb{E}\left[x_t \,|\mathcal{F}_t\right] \end{array}\right. \qquad t=1,\ldots,T.$$

3.4.3 Geographische Dekomposition

Wir nehmen nun an, daß das Problem (MSP) eine block-separable Struktur in der Form hat, daß Partitionierungen $\{I_{t,1},\ldots,I_{t,p}\}$ von $\{1,\ldots,r_t\}$, $t=1,\ldots,T$, existieren, so daß

$$\begin{aligned} f(x,\xi) &= \sum_{k=1}^{p} f_k(x_{I_k},\xi), \\ g_t(x_1,\ldots,x_t,\zeta_t) &= \begin{pmatrix} \sum_{k=1}^{p} g_{t,k}\left(x_{I_{1,k}},\ldots,x_{I_{t,k}},\zeta_t\right) \\ h_{t,1}\left(x_{I_{1,k}},\ldots,x_{I_{t,k}},\zeta_t\right) \\ \vdots \\ h_{t,k}\left(x_{I_{1,k}},\ldots,x_{I_{t,k}},\zeta_t\right) \end{pmatrix} \\ X_t(\zeta_t) &= X_{t,1}(\zeta_t) \times \cdots \times X_{t,p}(\zeta_t), \end{aligned}$$

$I_k := \bigcup_{t=1}^{T} I_{t,k}$, $x_{I_k} := \left(x_{I_{1,k}},\ldots,x_{I_{T,k}}\right)$, $f_k : \mathbb{R}^{|I_k|} \times \mathbb{R}^s \to \mathbb{R}$, $g_{t,k} : \mathbb{R}^{\sum_{\tau=1}^{t}|I_{\tau,k}|} \times \mathbb{R}^{\sum_{\tau=1}^{t} s_\tau} \to \mathbb{R}^{d_{t,0}}$, $h_{t,k} : \mathbb{R}^{\sum_{\tau=1}^{t}|I_{\tau,k}|} \times \mathbb{R}^{\sum_{\tau=1}^{t} s_\tau} \to \mathbb{R}^{d_{t,k}}$ und $X_{t,k} : \mathbb{R}^{\sum_{\tau=1}^{t} s_\tau} \rightrightarrows \mathbb{R}^{|I_{t,k}|}$, $k=1,\ldots,p$, $\sum_{k=0}^{p} d_{t,k} = d_t$, $t=1,\ldots,T$. Bei der Auftrennung von g_t unterscheiden wir hier zwischen einer Summe von Funktionen $g_{t,k}$ über k und Funktionen $h_{t,k}$, $k=1,\ldots,p$, um zeitkoppelnde Restriktionen innerhalb eines Blockes zu erlauben.

Wir führen wieder duale Parameter $\lambda_t \in L^1\left(\Omega, \mathcal{F}, \mathbb{P}; \mathbb{R}^{d_{t,0}}\right)$, $t = 1, \ldots, T$, ein. Dann ist die geographische Dekomposition definiert durch Dualisierung der $\sum_k g_{t,k}$-Restriktionen,

$$\begin{array}{ll} \max\limits_{\lambda=(\lambda_t)_{t=1}^T} & \operatorname{val}(D_\lambda) \\ \text{s.d.} & \lambda_t \geq 0 \quad \mathbb{P}\text{-f.s.} \quad t = 1, \ldots, T \end{array}$$

wobei das Lagrange-Problem

$$\begin{array}{lllll} \min\limits_x & L(x, \xi; \lambda) & & & \\ & x_{I_{t,k}} \in X_{t,k}(\zeta_t) & \mathbb{P}\text{-f.s.} & k = 1, \ldots, p & t = 1, \ldots, T \\ \text{s.d.} & h_{t,k}\left(x_{I_{1,k}}, \ldots, x_{I_{t,k}}, \zeta_t\right) \leq 0 & \mathbb{P}\text{-f.s.} & k = 1, \ldots, p & t = 1, \ldots, T \\ & x_{I_{t,k}} = \mathbb{E}\left[x_{I_{t,k}} | \mathcal{F}_t\right] & & k = 1, \ldots, p & t = 1, \ldots, T \end{array} \qquad (D_\lambda)$$

mit separabler Lagrangefunktion

$$\begin{aligned} L(x, \xi; \lambda) &:= \sum_{k=1}^p L_k\left(x_{I_k}, \xi; \lambda\right), \\ L_k\left(x_{I_k}, \xi; \lambda\right) &:= \mathbb{E}\left[f_k\left(x_{I_k}, \xi\right) + \sum_{t=1}^T \lambda_t^T g_{t,k}\left(x_{I_{1,k}}, \ldots, x_{I_{t,k}}, \zeta_t\right)\right], \qquad k = 1, \ldots, p, \end{aligned}$$

in partielle Lagrangeprobleme

$$\begin{array}{llll} \min\limits_{x_{I_k}} & L_k\left(x_{I_k}, \xi; \lambda\right) & & \\ & x_{I_{t,k}} \in X_{t,k}(\zeta_t) & \mathbb{P}\text{-f.s.} & t = 1, \ldots, T \\ \text{s.d.} & h_{t,k}\left(x_{I_{1,k}}, \ldots, x_{I_{t,k}}, \zeta_t\right) \leq 0 & \mathbb{P}\text{-f.s.} & t = 1, \ldots, T \\ & x_{I_{t,k}} = \mathbb{E}\left[x_{I_{t,k}} | \mathcal{F}_t\right] & & t = 1, \ldots, T \end{array} \qquad (D_{\lambda,k})$$

zerfällt. Für die aggregierte Formulierung (MSP$[\mathcal{N}]$) definieren wir ähnlich die Lagrangefunktion

$$\begin{aligned} L_{\mathcal{N}}(x, \xi; \lambda) &:= \sum_{k=1}^p L_{k,\mathcal{N}}\left(x_{I_k}, \xi; \lambda\right), \\ L_{k,\mathcal{N}}\left(x_{I_k}, \xi; \lambda\right) &:= \sum_{N \in \mathcal{N}}\left(\left(\sum_{n \in N} \pi_n f_k\left(x_{I_k}^{p(N)}, \zeta^n\right)\right) + \pi_N \sum_{t=1}^T \lambda_{n_t(N)}^T g_{t,k}\left(x_{I_k}^{p(n_t(N))}, \zeta^{n_t(N)}\right)\right) \end{aligned}$$

mit dualen Parametern $(\lambda_n)_{n \in \bar{N}|_{\mathcal{N}}}$, so daß wir die partiellen Lagrangeprobleme

$$\begin{array}{lll} \min\limits_{x_{I_k}} & L_{k,\mathcal{N}}\left(x_{I_k}, \xi; \lambda\right) & \\ \text{s.d.} & x_{I_{t(n),k}}^n \in X_{t(n),k}(\zeta^n) & n \in \bar{N}|_{\mathcal{N}} \\ & h_{t(n),k}\left(x_{I_k}^{p(n)}, \zeta^n\right) \leq 0 & n \in \bar{N}|_{\mathcal{N}} \end{array} \qquad (D_{\lambda,k}[\mathcal{N}])$$

und das duale Problem

$$\max_{\lambda \geq 0} \quad \sum_{k=1}^p \operatorname{val}\left(D_{\lambda,k}[\mathcal{N}]\right) \qquad (D[\mathcal{N}])$$

erhalten. Für einen Lösungspunkt $\hat{x}_{I_k}$ von $(D_{\lambda,k}[\mathcal{N}])$ ist $L_{k,\mathcal{N}}(\cdot; \lambda) \geq L_{k,\mathcal{N}}(\hat{x}_{I_k}; \lambda)$ ein zulässiger Lagrange-Schnitt, welcher zu $\mathcal{C}_{\mathcal{N}}$ hinzugefügt wird.

Verfeinerungsoperationen

Wie bereits für die abstrakte Formulierung aufgezeigt, ist es wünschenswert, für eine Verfeinerung $\mathcal{N}' \preceq \mathcal{N}$ einen Schnitt aus $\mathcal{C}_{\mathcal{N}}$ nach $\mathcal{C}_{\mathcal{N}'}$ zu übernehmen. Dies kann man erreichen durch "Umformung" von $\hat{x}_{I_k}$ zu einem Lösungspunkt von $(D_{\lambda',k}[\mathcal{N}'])$, wobei $\lambda'^n := \lambda^{\text{agg}(n)}$ für $n \in \bar{N}|_{\mathcal{N}'}$ und $\text{agg} : \bar{N} \to \bar{N}|_{\mathcal{N}}$ ist die Abbildung, die die Aggregierung $\mathcal{N}$ ausdrückt (es galt $\bar{N}|_{\mathcal{N}'} \subseteq \bar{N}|_{\mathcal{N}}$). Daher definieren wir den Operator $E_k : X_{k,\mathcal{N}} \to X_{k,\mathcal{N}'}$ mittels

$$E_k(\hat{x}_{I_k}) := \operatorname{argmin}\left((D_{\lambda',k}[\mathcal{N}']); \hat{x}'_{I_k}\right)$$

wobei $\hat{x}'^n_{I_k} := \hat{x}^{\text{agg}(n)}_{I_k}$, $n \in \bar{N}|_{\mathcal{N}'}$, den Startpunkt zur Lösung von $(D_{\lambda',k}[\mathcal{N}'])$ bezeichne. Wenn die Probleme $(D_{\lambda,k}[\mathcal{N}])$ und $(D_{\lambda',k}[\mathcal{N}'])$ sich nicht zu stark unterscheiden, kann der Punkt $\hat{x}'_{I_k}$ schon in der Nähe eines Lösungspunktes von $(D_{\lambda',k}[\mathcal{N}'])$ liegen, so daß die Lösung von $(D_{\lambda',k}[\mathcal{N}'])$ nicht zu teuer wird. Der aktualisierte Lagrange-Schnitt $L_{k,\mathcal{N}'}(\cdot_{I_k}, \xi; \lambda') \geq L_{k,\mathcal{N}'}(E_k(\hat{x}_{I_k}), \xi; \lambda')$ kann dann zu $\mathcal{C}_{\mathcal{N}'}$ hinzugefügt werden.

Ähnlich kann die Aktualisierung der Extrempunkte $Z_{k,\mathcal{N}}$ erreicht werden mit $Z_{k,\mathcal{N}'} := E_k(Z_{k,\mathcal{N}})$, aber da die Optimalität der Extrempunkte $Z_{k,\mathcal{N}}$ nicht notwendig war, kann der Lösungsprozess von $(D_{\lambda',k}[\mathcal{N}'])$, der in einem Punkt von $Z_{k,\mathcal{N}}$ gestartet wird, gestoppt werden, sobald ein zulässiger Punkt erreicht wird.

3.5 Adaptiver Szenarioreduktionsalgorithmus

Wir sind nun fast so weit, daß wir den adaptiven Diskretisierungsalgorithmus 2.2 für mehrstufige stochastische Programme formulieren können. Die letzten fehlenden Bestandteile sind eine angepasste Formel (2.1) und die Formulierung des Problemes (Proj$[\mathcal{N},\mathcal{N}']$) für MSPs.

Die Funktion von (Proj$[\mathcal{N},\mathcal{N}']$) ist die Minimierung des Abstandes zwischen einem Entscheidungsprozess auf $\bar{N}|_{\mathcal{N}'}$ zu einem entsprechenden Prozess auf $\bar{N}|_{\mathcal{N}}$, falls $\text{val}(\text{CMSP}[\mathcal{N}]) - \text{val}(\text{CMSP}[\mathcal{N}']) > \sigma_P |\text{val}(\text{CMSP}[\mathcal{N}'])|$. Daher definieren wir

$$\begin{array}{lll}
\min\limits_{x} & \sum\limits_{N\in\mathcal{N}} \pi_N \max\limits_{t=1,\dots,T} \max\limits_{N'\in\mathcal{N}'|_N} \left\| x^{n_t(N')} - x^{n_t(N)} \right\|_\infty & \\
 & \check{g}_{t(n)}\left(x^{p(n)}, \zeta^n\right) \leq 0 & n \in \bar{N}|_{\mathcal{N}'} \\
\text{s.d.} & x^n \in \check{X}_{t(n)}(\zeta^n) & n \in \bar{N}|_{\mathcal{N}'} \\
 & c(x) \leq 0 & c \in \mathcal{C}_{\mathcal{N}'} \\
 & \sum\limits_{N'\in\mathcal{N}'} \sum\limits_{n\in N'} \pi_n \check{f}\left(x^{p(N')}, \zeta^n\right) \leq \text{val}(\text{CMSP}[\mathcal{N}']) + \sigma_P \cdot |\text{val}(\text{CMSP}[\mathcal{N}'])| &
\end{array} \tag{ProjMSP$[\mathcal{N},\mathcal{N}']$}$$

als Formulierung von (Proj$[\mathcal{N},\mathcal{N}']$) für MSPs.

Im Fall $\text{val}(C[\mathcal{N}']) - \text{val}(C[\mathcal{N}]) > \sigma_P |\text{val}(C[\mathcal{N}])|$ hat Algorithmus 2.2 eine Menge $\mathcal{M} \subseteq \mathcal{N}$ ausgewählt, so daß die Aggregierungen $M \in \mathcal{M}$ einen hohen (positiven) Anteil an der Differenz $\text{val}(C[\mathcal{N}']) - \text{val}(C[\mathcal{N}])$ haben. Für MSPs können wir die Formel (2.1) mittels

$$\mathcal{M} := \operatorname{argmax}\left\{ \sum_{M\in\mathcal{M}} \sum_{M'\in\mathcal{N}'|_M} \pi_{M'} \sum_{n\in M'} \left(\check{f}\left(x'^{p(M')}, \zeta^n\right) - \check{f}\left(x^{p(M)}, \zeta^n\right)\right) \middle| \mathcal{M} \subseteq \mathcal{N}, |\mathcal{M}| \leq \bar{m} \right\}. \tag{3.18}$$

für $x \in \text{Argmin}(\text{CMSP}[\mathcal{N}])$ und $x' \in \text{Argmin}(\text{CMSP}[\mathcal{N}'])$ anpassen. Wegen $\text{val}(\text{CMSP}[\mathcal{N}']) > \text{val}(\text{CMSP}[\mathcal{N}])$ wird $\mathcal{M}$ nicht leer sein.

Wir können nun Algorithmus 2.2 für mehrstufige stochastische Programme angeben. Der Parameter σ_P kontrolliert, wie weit es dem Quotienten $\frac{\text{val}(\text{CMSP}[\mathcal{N}])}{\text{val}(\text{CMSP}[\mathcal{N}'])}$ erlaubt ist, sich von 1 zu entfernen.

σ_D kontrolliert, wie nah $\bar{N}|_{\mathcal{N}'}$ und $\bar{N}$ am Ende sein müssen, relativ zum Abstand zwischen der anfänglichen Aggregierung und $\bar{N}$. Der Parameter $\bar{m}$ gibt eine obere Schranke an die Größe von $\mathcal{M}$ an, wenn Formel (3.18) oder (2.2) angewandt wird, und s kontrolliert grob die Anzahl neuer Szenarien, die während einer Verfeinerung zu $\mathcal{N}'$ hinzugefügt werden.

Algorithmus 3.11 (Adaptive Szenarioreduktion).
Input: Anfänglicher Szenariobaum $\bar{N}$, Parameter $\sigma_P \geq 0$, $\sigma_D \in [0,1]$, $\bar{m} \in \mathbb{N}$, $s \in \mathbb{N}$, $s \geq \bar{m}$
Compute:

Berechne eine anfängliche Aggregierung $\mathcal{N} := \{\bar{N}_T\}$ durch Auswahl eines Szenarios mittels forward selection (vgl. 3.3.2). $\mathcal{D}_0 := \zeta_c\left(\mathbb{P}, \delta_{\omega_{i(\bar{N}_T)}}\right) = p_{i(\bar{N}_T)} \min_{n \in \bar{N}_T \setminus \{n_T(\bar{N}_T)\}} c\left(\zeta^{n_T(\bar{N}_T)}, \zeta^n\right)$ sei der Abstand zwischen dem anfänglichen Szenariobaum und dem Baum, der nur aus dem ausgewählten Szenario besteht.
Verfeinere $\mathcal{N}$ zu einer Aggregierung $\mathcal{N}'$ mittels Algorithmus 3.6 mit $s_{\bar{N}_T} := s$.
Initialisiere $Z_{k,\mathcal{N}}$, $\mathcal{C}_{\mathcal{N}} := \emptyset$.
repeat
  Aktualisiere $\mathcal{C}_{\mathcal{N}}$ und berechne die Schnitte $\mathcal{C}_{\mathcal{N}'}$ aus $\mathcal{C}_{\mathcal{N}}$.
  Löse (CMSP[$\mathcal{N}$]) und (CMSP [$\mathcal{N}'$]).
  if val (CMSP [$\mathcal{N}'$]) $-$ val (CMSP[$\mathcal{N}$]) $> \sigma_P$ |val (CMSP[$\mathcal{N}$])| **then**
    Berechne $\mathcal{M}$ entsprechend (3.18) und setze $s_M := [s \cdot \rho_M]$, wobei ρ_M der Bruch aus Formel (2.3) ist, $M \in \mathcal{M}$.
    Berechne $\mathcal{N}(\mathcal{M})$ und $\mathcal{N}'(\mathcal{M})$ mittels Algorithmus 3.6.
    $Z_{k,\mathcal{N}(\mathcal{M})} := E_k(Z_{k,\mathcal{N}})$, $k = 1, \ldots, p$.
    $\mathcal{N} := \mathcal{N}(\mathcal{M})$, $\mathcal{N}' := \mathcal{N}'(\mathcal{M})$
  else if val (CMSP[$\mathcal{N}$]) $-$ val (CMSP [$\mathcal{N}'$]) $> \sigma_P$ |val (CMSP [$\mathcal{N}'$])| **then**
    Löse (ProjMSP[$\mathcal{N}, \mathcal{N}'$]).
    Berechne $\mathcal{M}$ entsprechend (2.2) und setze $s_M := [s \cdot \rho_M]$, wobei ρ_M der Bruch aus Formel (2.3) ist, $M \in \mathcal{M}$.
    Berechne $\mathcal{N}(\mathcal{M})$ und $\mathcal{N}'(\mathcal{M})$ mittels Algorithmus 3.6.
    $Z_{k,\mathcal{N}(\mathcal{M})} := E_k(Z_{k,\mathcal{N}})$, $k = 1, \ldots, p$.
    $\mathcal{N} := \mathcal{N}(\mathcal{M})$, $\mathcal{N}' := \mathcal{N}'(\mathcal{M})$
  else if $\tilde{\mathcal{D}}(\mathcal{N}') > \sigma_D \mathcal{D}_0$ **then**
    Setze $\mathcal{M} := \operatorname{argmax}\left\{\sum_{M \in \mathcal{M}} \tilde{\mathcal{D}}(M) \,\middle|\, \mathcal{M} \subseteq \mathcal{N},\, |\mathcal{M}| \leq \bar{m},\, \tilde{\mathcal{D}}(M) > 0, M \in \mathcal{M}\right\}$.
    Bestimme $\mathcal{N}'(\mathcal{M})$ mittels Algorithmus 3.6, $s_M := s$ für $M \in \mathcal{M}$.
    $\mathcal{N}' := \mathcal{N}'(\mathcal{M})$ ($\mathcal{N}$ bleibt unverändert)
  end if
until $\mathcal{M} = \emptyset$
Output: Reduzierter Szenariobaum $\bar{N}|_{\mathcal{N}}$.

3.6 Gemischt-ganzzahlige lineare mehrstufige stochastische Programme

Der adaptive Szenarioreduktionsalgorithmus 3.11 wurde für lineare mehrstufige stochastische Programme implementiert und wird hoffentlich auf den Fall erweitert, der ganzzahlige Variablen enthält. Daher fahren wir damit fort, die Notation für gemischt-ganzzahlige lineare mehrstufige stochastische Programme einzuführen. Dann erwähnen wir noch eine alternative Möglichkeit, das aggregierte Problem zu definieren. Als ein Beispiel für ein lineares MSP beschreiben wir ein Modell für das Strommanagement in einem hydro-thermalen System unter unsicherer Last. Wir schließen diesen Abschnitt ab mit einem Schnittebenenverfahren für die Lösung linearer MSPs, welches auf der Relaxierung der zeitkoppelnden Restriktionen basiert.

Das mehrstufige stochastische Programm (MSP) ist linear, wenn die Funktionen $f, g_t, t = 1, \ldots, T$, linear sind in x und $X_t(\zeta_t)$ polyhedral ist in $\mathbb{R}^{r_t}$ für alle $\zeta_t \in \zeta_t(\Omega)$ und $t = 1, \ldots, T$. Daher, wenn wir die geographische Dekomposition von (MSP) im Kopf behalten, nehmen wir an, daß

$$\begin{aligned} f(x,\xi) &= \sum_{k=1}^{p} \left\langle c_k(\xi), x_{I_k} \right\rangle, \\ g_t(x_1,\dots,x_t;\zeta_t) &= \begin{pmatrix} \left(\sum_{\tau=1}^{t} \sum_{k=1}^{p} A_{t,\tau,k}(\zeta_t)\, x_{I_{\tau,k}} \right) - b_t(\zeta_t) \\ \sum_{\tau=1}^{t} \tilde{A}_{t,\tau,1}(\zeta_t)\, x_{I_{\tau,1}} - \tilde{b}_{t,1}(\zeta_t) \\ \vdots \\ \sum_{\tau=1}^{t} \tilde{A}_{t,\tau,p}(\zeta_t)\, x_{I_{\tau,p}} - \tilde{b}_{t,p}(\zeta_t) \end{pmatrix}, \\ X_t(\zeta_t) &= X_{t,1}(\zeta_t) \times \cdots \times X_{t,k}(\zeta_t) \end{aligned}$$

wobei $c_k : \mathbb{R}^{\sum_{\tau=1}^{T} s_\tau} \to \mathbb{R}^{|I_k|}$, $A_{t,\tau,k} : \mathbb{R}^{\sum_{\tau'=1}^{t} s_{\tau'}} \to \mathbb{R}^{d_{t,0} \times |I_{\tau,k}|}$, $\tau = 1,\dots,t$, $b_t : \mathbb{R}^{\sum_{\tau=1}^{t} s_\tau} \to \mathbb{R}^{d_{t,0}}$, $\tilde{A}_{t,\tau,k} : \mathbb{R}^{\sum_{\tau'=1}^{t} s_{\tau'}} \to \mathbb{R}^{d_{t,k} \times |I_{\tau,k}|}$, $\tau = 1,\dots,t$, $\tilde{b}_{t,k} : \mathbb{R}^{\sum_{\tau'=1}^{t} s_{\tau'}} \to \mathbb{R}^{d_{t,k}}$ und $X_{t,k}(\zeta_t) \subseteq \mathbb{R}^{|I_{t,k}|}$ polyhedral ist für alle $\zeta_t \in \zeta_t(\Omega)$ und $t = 1,\dots,T$.

Ein gemischt-ganzzahliges lineares (MSP) liest sich nun

$$\begin{array}{lllll} \min\limits_{x} & \mathbb{E}\left[\sum_{k=1}^{p} \left\langle c_k(\xi), x_{I_k} \right\rangle \right] & & & \\ & \sum_{\tau=1}^{t} \sum_{k=1}^{p} A_{t,\tau,k}(\zeta_t)\, x_{I_{\tau,k}} \le b_t(\zeta_t) & & \mathbb{P}\text{-f.s.} & t = 1,\dots,T \\ \text{s.d.} & \sum_{\tau=1}^{t} \tilde{A}_{t,\tau,k}(\zeta_t)\, x_{I_{\tau,k}} \le \tilde{b}_{t,k}(\zeta_t) & \mathbb{P}\text{-f.s.} & k = 1,\dots,p & t = 1,\dots,T \\ & x_{I_{t,k}} \in X_{t,k}(\zeta_t) & \mathbb{P}\text{-f.s.} & k = 1,\dots,p & t = 1,\dots,T \\ & x_{B_{t,k}} \in \mathbb{Z}^{|B_{t,k}|} & \mathbb{P}\text{-f.s.} & k = 1,\dots,p & t = 1,\dots,T \\ & x_{I_{t,k}} = \mathbb{E}\left[x_{I_{t,k}} | \mathcal{F}_t \right] & & k = 1,\dots,p & t = 1,\dots,T \end{array} \qquad \text{(MILMSP)}$$

wobei $B_{t,k} \subseteq I_{t,k}$ die Menge der Variablen mit Ganzzahligkeitsrestriktionen bezeichne, $k = 1,\dots,p$, $t = 1,\dots,T$.

Die Lagrangefunktion, die man durch Dualisierung der koppelnden Restriktionen $\sum\sum A(\zeta)\, x \le b(\zeta)$ erhält, ist

$$\begin{aligned} L(x,\xi;\lambda) &= \sum_{k=1}^{p} L_k(x_{I_k},\xi;\lambda) - \mathbb{E}\left[\sum_{t=1}^{T} \lambda_t^T b_t(\zeta_t) \right], \\ L_k(x_{I_k},\xi;\lambda) &= \mathbb{E}\left[\left\langle c_k(\xi), x_{I_k} \right\rangle + \sum_{t=1}^{T} \lambda_t^T \sum_{\tau=1}^{t} A_{t,\tau,k}(\zeta_t)\, x_{I_{\tau,k}} \right], \end{aligned}$$

wobei $\lambda_t \in \mathbb{R}^{d_{t,0}}$. Die partiellen Lagrangeprobleme sind daher

$$\begin{array}{llll} \min\limits_{x_{I_k}} & L_k(x_{I_k},\xi;\lambda) & & \\ & \sum_{\tau=1}^{t} \tilde{A}_{t,\tau,k}(\zeta_t)\, x_{I_{\tau,k}} \le \tilde{b}_{t,k}(\zeta_t) & \mathbb{P}\text{-f.s.} & t = 1,\dots,T \\ \text{s.d.} & x_{I_{t,k}} \in X_{t,k}(\zeta_t) & \mathbb{P}\text{-f.s.} & t = 1,\dots,T \\ & x_{B_{t,k}} \in \mathbb{Z}^{|B_{t,k}|} & \mathbb{P}\text{-f.s.} & t = 1,\dots,T \\ & x_{I_{t,k}} = \mathbb{E}\left[x_{I_{t,k}} | \mathcal{F}_t \right] & & t = 1,\dots,T \end{array}$$

3.6.1 Diskretisierung, Aggregierung

Ähnlich zu allgemeinen MSPs erhalten wir die Szenariobaumformulierung

$$\begin{array}{lllll}
\min\limits_{x} & \sum\limits_{k=1}^{p}\sum\limits_{n\in\bar{N}_T}\pi_n\left\langle c_k\left(\zeta^n\right),x_{I_k}^{p(n)}\right\rangle & & & \\
& \sum\limits_{\tau=1}^{t(n)}\sum\limits_{k=1}^{p}A_{t(n),\tau,k}\left(\zeta^n\right)x_{I_{\tau,k}}^{n-t(n)+\tau}\leq b_{t(n)}\left(\zeta^n\right) & & n\in\bar{N} & \\
\text{s.d.} & \sum\limits_{\tau=1}^{t(n)}\tilde{A}_{t(n),\tau,k}\left(\zeta^n\right)x_{I_{\tau,k}}^{n-t(n)+\tau}\leq\tilde{b}_{t(n),k}\left(\zeta^n\right) & k=1,\ldots,p & n\in\bar{N} & (\text{MILMSP}\left[\bar{N}\right])\\
& x_{I_{t(n),k}}^{n}\in X_{t(n),k}\left(\zeta^n\right) & k=1,\ldots,p & n\in\bar{N} & \\
& x_{B_{t(n),k}}^{n}\in\mathbb{Z}^{\left|B_{t(n),k}\right|} & k=1,\ldots,p & n\in\bar{N} &
\end{array}$$

und die aggregierte Formulierung

$$\begin{array}{lllll}
\min\limits_{x} & \sum\limits_{k=1}^{p}\sum\limits_{N\in\mathcal{N}}\sum\limits_{n\in N}\pi_n\left\langle c_k\left(\zeta^n\right),x_{I_k}^{p(N)}\right\rangle & & & \\
& \sum\limits_{\tau=1}^{t(n)}\sum\limits_{k=1}^{p}A_{t(n),\tau,k}\left(\zeta^n\right)x_{I_{\tau,k}}^{n-t(n)+\tau}\leq b_{t(n)}\left(\zeta^n\right) & & n\in\bar{N}\left|_{\mathcal{N}}\right. & \\
\text{s.d.} & \sum\limits_{\tau=1}^{t(n)}\tilde{A}_{t(n),\tau,k}\left(\zeta^n\right)x_{I_{\tau,k}}^{n-t(n)+\tau}\leq\tilde{b}_{t(n),k}\left(\zeta^n\right) & k=1,\ldots,p & n\in\bar{N}\left|_{\mathcal{N}}\right. & (\text{MILMSP}[\mathcal{N}])\\
& x_{I_{t(n),k}}^{n}\in X_{t(n),k}\left(\zeta^n\right) & k=1,\ldots,p & n\in\bar{N}\left|_{\mathcal{N}}\right. & \\
& x_{B_{t(n),k}}^{n}\in\mathbb{Z}^{\left|B_{t(n),k}\right|} & k=1,\ldots,p & n\in\bar{N}\left|_{\mathcal{N}}\right. &
\end{array}$$

Die Lagrangefunktion von (MILMSP[$\mathcal{N}$]) ist

$$\begin{aligned}
L_{\mathcal{N}}\left(x,\xi;\lambda\right) &:= \sum_{k=1}^{p}L_{\mathcal{N},k}\left(x_{I_k},\xi;\lambda\right)-\sum_{N\in\mathcal{N}}\pi_N\sum_{t=1}^{T}\lambda_{n_t(N)}^{T}b_t\left(\zeta^{n_t(N)}\right),\\
L_{\mathcal{N},k}\left(x_{I_k},\xi;\lambda\right) &:= \sum_{N\in\mathcal{N}}\left\langle\sum_{n\in\mathcal{N}}\pi_n c_k\left(\zeta^n\right),x_{I_k}^{p(N)}\right\rangle+\sum_{N\in\mathcal{N}}\pi_N\sum_{t=1}^{T}\lambda_{n_t(N)}^{T}\sum_{\tau=1}^{t}A_{t,\tau,k}\left(\zeta^{n_t(N)}\right)x_{I_{t,k}}^{n_\tau(N)}
\end{aligned}$$

und die partiellen Lagrangeprobleme lesen sich

$$\begin{array}{llll}
\min\limits_{x_{I_k}} & L_{\mathcal{N},k}\left(x_{I_k},\xi;\lambda\right) & & \\
& \sum\limits_{\tau=1}^{t(n)}\tilde{A}_{t(n),\tau,k}\left(\zeta^n\right)x_{I_{\tau,k}}^{n-t(n)+\tau}\leq\tilde{b}_{t(n),k}\left(\zeta^n\right) & n\in\bar{N}\left|_{\mathcal{N}}\right. & \\
\text{s.d.} & x_{I_{t(n),k}}^{n}\in X_{t(n),k}\left(\zeta^n\right) & n\in\bar{N}\left|_{\mathcal{N}}\right. & k=1,\ldots,p.\\
& x_{B_{t(n),k}}^{n}\in\mathbb{Z}^{\left|B_{t(n),k}\right|} & n\in\bar{N}\left|_{\mathcal{N}}\right. &
\end{array}$$

Eine konvexe Relaxierung von (MILMSP) definiert sich durch Weglassen der Ganzzahligkeitsbedingungen und durch Hinzufügung von linearen Schnitten. Wir identifizieren die Menge der Schnitte $\mathcal{C}_{\mathcal{N}}$ hier mit Paaren $\left(\{a^n\}_{n\in\bar{N}|_{\mathcal{N}}},\bar{a}\right)\in\left(\times_{n\in\bar{N}|_{\mathcal{N}}}\mathbb{R}^{r_{t(n)}}\right)\times\mathbb{R}$, so daß ein Schnitt definiert ist mittels $\sum_{n\in\bar{N}|_{\mathcal{N}}}\langle a^n,x^n\rangle\le\bar{a}$. Dann liest sich die Konvexifizierung von (MILMSP[$\mathcal{N}$]) als

$$\begin{array}{llll}
\min\limits_{x} & \sum\limits_{k=1}^{p}\sum\limits_{N\in\mathcal{N}}\sum\limits_{n\in N}\pi_n\left\langle c_k\left(\zeta^n\right),x_{I_k}^{p(N)}\right\rangle & & \\
& \sum\limits_{\tau=1}^{t(n)}\sum\limits_{k=1}^{p}A_{t(n),\tau,k}\left(\zeta^n\right)x_{I_{\tau,k}}^{n-t(n)+\tau}\le b_{t(n)}\left(\zeta^n\right) & & n\in\bar{N}|_{\mathcal{N}} \\
\text{s.d.} & \sum\limits_{\tau=1}^{t(n)}\tilde{A}_{t(n),\tau,k}\left(\zeta^n\right)x_{I_{\tau,k}}^{n-t(n)+\tau}\le\tilde{b}_{t(n),k}\left(\zeta^n\right) & k=1,\ldots,p & n\in\bar{N}|_{\mathcal{N}} \\
& x_{I_{t(n),k}}^{n}\in X_{t(n),k}\left(\zeta^n\right) & k=1,\ldots,p & n\in\bar{N}|_{\mathcal{N}} \\
& \sum\limits_{n\in\bar{N}|_{\mathcal{N}}}\langle a^n,x^n\rangle\le\bar{a} & & \left(\{a^n\}_{n\in\bar{N}|_{\mathcal{N}}},\bar{a}\right)\in\mathcal{C}_{\mathcal{N}}
\end{array}\qquad(\text{LMSP}[\mathcal{N}])$$

3.6.2 Aggregierung: 2. Möglichkeit

In den Arbeiten [1] und [8] wurde eine andere Methode zur Aggregierung bei linearen mehrstufigen stochastischen Programmen verwendet. Anstatt für jedes $N\in\mathcal{N}$ ein Szenario auszuwählen und das MSP zu betrachten, welches durch $\bar{N}|_{\mathcal{N}}$ induziert wird, wurde dort nicht nur die Zielfunktion gemittelt, wie wir das getan haben, sondern auch die Restriktionen. Dies erlaubt die Berechnung von unteren und oberen Schranken an $\operatorname{val}\left(\text{MSP}\left[\bar{N}\right]\right)$, aber stellt auch zusätzliche Annahmen an den zugrundeliegenden stochastischen Prozess, da die $\mathbb{P}$-f.s.-Struktur der Restriktionen kein beliebiges Mitteln erlauben muss. Zur Vollständigkeit führen wir hier kurz die Notation für diese Art der Aggregierung ein:

Angenommen, der Polyeder $X_{t,k}\left(\zeta_t\right)$ ist in der Form

$$X_{t,k}\left(\zeta_t\right)=\left\{x_{I_{t,k}}\in\mathbb{R}^{|I_{t,k}|}\,\middle|\,\bar{A}_{t,k}\left(\zeta_t\right)x_{I_{t,k}}\le\bar{b}_{t,k}\left(\zeta_t\right)\right\}$$

gegeben, so daß (MILMSP) die Form

$$\begin{array}{lllll}
\min\limits_{x} & \mathbb{E}\left[\sum\limits_{k=1}^{p}\left\langle c_k\left(\xi\right),x_{I_k}\right\rangle\right] & & & \\
& \sum\limits_{\tau=1}^{t}\sum\limits_{k=1}^{p}A_{t,\tau,k}\left(\zeta_t\right)x_{I_{\tau,k}}\le b_t\left(\zeta_t\right) & \mathbb{P}\text{-f.s.} & & t=1,\ldots,T \\
\text{s.d.} & \sum\limits_{\tau=1}^{t}\tilde{A}_{t,\tau,k}\left(\zeta_t\right)x_{I_{\tau,k}}\le\tilde{b}_{t,k}\left(\zeta_t\right) & \mathbb{P}\text{-f.s.} & k=1,\ldots,p & t=1,\ldots,T \\
& \bar{A}_{t,k}\left(\zeta_t\right)x_{I_{t,k}}\le\bar{b}_{t,k}\left(\zeta_t\right) & \mathbb{P}\text{-f.s.} & k=1,\ldots,p & t=1,\ldots,T \\
& x_{B_{t,k}}\in\mathbb{Z}^{|B_{t,k}|} & \mathbb{P}\text{-f.s.} & k=1,\ldots,p & t=1,\ldots,T \\
& x_{I_{t,k}}=\mathbb{E}\left[x_{I_{t,k}}\,|\mathcal{F}_t\right] & & k=1,\ldots,p & t=1,\ldots,T
\end{array}$$

hat. Dann ist eine andere Möglichkeit, daß aggregierte Problem zu definieren, die folgende:

$$\min_x \quad \sum_{k=1}^{p} \sum_{N\in\mathcal{N}} \left\langle \sum_{n\in N} \pi_n c_k\left(\zeta^n\right), x_{I_k}^{p(N)} \right\rangle$$

$$\text{s.d.} \quad \sum_{\tau=1}^{t(n)} \sum_{k=1}^{p} \left(\sum_{n'\in \text{agg}^{-1}(n)} \pi_{n'} A_{t(n),\tau,k}\left(\zeta^{n'}\right)\right) x_{I_{\tau,k}}^{n-t(n)+\tau} \le \sum_{n'\in \text{agg}^{-1}(n)} \pi_{n'} b_{t(n)}\left(\zeta^{n'}\right) \qquad n \in \bar{N}\,|_{\mathcal{N}}$$

$$\sum_{\tau=1}^{t(n)} \left(\sum_{n'\in \text{agg}^{-1}(n)} \pi_{n'} \tilde{A}_{t(n),\tau,k}\left(\zeta^{n'}\right)\right) x_{I_{\tau,k}}^{n-t(n)+\tau} \le \sum_{n'\in \text{agg}^{-1}(n)} \pi_{n'} \tilde{b}_{t(n)}\left(\zeta^{n'}\right) \qquad k=1,\dots,p \quad n \in \bar{N}\,|_{\mathcal{N}}$$

$$\left(\sum_{n'\in \text{agg}^{-1}(n)} \pi_{n'} \bar{A}_{t(n),k}\left(\zeta^{n'}\right)\right) x_{I_{t(n),k}}^{n} \le \sum_{n'\in \text{agg}^{-1}(n)} \pi_{n'} \bar{b}_{t(n),k}\left(\zeta^{n'}\right) \qquad k=1,\dots,p \quad n \in \bar{N}\,|_{\mathcal{N}}$$

$$x_{B_{t(n),k}}^{n} \in \mathbb{Z}^{\left|B_{t(n),k}\right|} \qquad k=1,\dots,p \quad n \in \bar{N}\,|_{\mathcal{N}}$$

3.6.3 Ein Beispiel: Strommanagement in einem hydro-thermalen System unter Unsicherheit

Nach der Einführung des Szenarioreduktionsalgorithmusses und all seiner Notation ist es nun an der Zeit, ein kurzes Beispiel für lineare mehrstufige stochastische Programme anzugeben. Die numerischen Ergebnisse im nächsten Kapitel basieren auf diesem Beispiel.

Wir betrachten ein Stromerzeugungssystem, welches aus einer Sammlung von kohlebetriebenen Wärmekraftwerken I und Pumpspeicherwerken J besteht. Das Ziel besteht darin, kostenoptimale Betriebsstufen für die Einheiten für eine Woche unter Last zu finden. Die elektrische Last unterliegt dabei Unsicherheiten.

Fixierte Parameter für die Wärmekraftwerke sind dabei die Brennstoffkosten $b \ge 0$ und Leistungsschranken $0 \le \underline{p} \le \bar{p}$. Die Parameter für die Pumpspeicherwerke sind die Effizienz der Pumpen $0 \le \eta \le 1$, die maximalen Betriebsstufen der Pumpen und Turbinen $\bar{w}, \bar{v} \ge 0$ und die Kapazität der Speicher $\bar{l} \ge 0$. Für den Beginn und das Ende der Woche sind die Füllstände l_{in} und l_{end} der Speicher vorgegeben .

Die stochastischen Daten bestehen aus der Last d und der verlangten Reserve r für eine Woche. Wir suchen nach einem Prozess von Entscheidung über die Betriebsstufen der Wärmekraftwerke p, der Pumpen w und Turbinen v, und der Füllstände l in den Pumpspeichern.

Der Entscheidungsprozess unterliegt den folgenden Restriktionen:

- Bedingungen an die Füllstände zu Beginn und zum Ende der Woche: $l_{j,1} = l_{\text{in},j} - (v_{j,1} - \eta_j w_{j,1})$, $l_{j,T} = l_{\text{end},j}$, $j \in J$
- Zeitkopplung der Füllstände: $l_{j,t} = l_{j,t-1} - (v_{j,t} - \eta_j w_{j,t})$, $j \in J$, $t = 2, \dots, T$
- Erfüllung der Last: $\sum_{i\in I} p_{i,t} + \sum_{j\in J} (v_{j,t} - w_{j,t}) \ge d_t$, $t = 1, \dots, T$
- Reserve-Forderung: $\sum_{i\in I} \bar{p}_i - p_{i,t} \ge r_t$, $t = 1, \dots, T$
- verschiedene Schranken: $\underline{p}_t \le p_{i,t} \le \bar{p}_t$, $i \in I$, $0 \le v_{j,t} \le \bar{v}_j$, $0 \le w_{j,t} \le \bar{w}_j$, $0 \le l_{j,t} \le \bar{l}_j$, $j \in J$, $t = 1, \dots, T$

Das Ziel ist die Minimierung der erwarteten Brennstoffkosten $\mathbb{E}\left[\sum_{i\in I}\sum_{t=1,\ldots,T} b_i p_{i,t}\right]$.

Das komplette Modell hat daher folgende Form

$$
\begin{array}{lrll}
\min\limits_{p,l,v,w} & \mathbb{E}\left[\sum_{i\in I}\sum_{t=1}^{T} b_i p_{i,t}\right] & & \\
& l_{j,1}+(v_{j,1}-\eta_j w_{j,1}) = l_{\mathrm{in},j} & & j\in J \\
& l_{j,T} = l_{\mathrm{end},j} & & j\in J \\
& l_{j,t}-l_{j,t-1}+(v_{j,t}-\eta_j w_{j,t}) = 0 & t=2,\ldots,T, & j\in J \\
& \sum_{i\in I} p_{i,t}+\sum_{j\in J}(v_{j,t}-w_{j,t}) \geq d_t & t=1,\ldots,T & \\
\text{s.d.} & \sum_{i\in I}\bar{p}_i - p_{i,t} \geq r_t & t=1,\ldots,T & \\
& \underline{p}_t \leq p_{i,t} \leq \bar{p}_t & t=1,\ldots,T & i\in I \\
& 0 \leq v_{j,t} \leq \bar{v}_j & t=1,\ldots,T & j\in J \\
& 0 \leq w_{j,t} \leq \bar{w}_j & t=1,\ldots,T & j\in J \\
& 0 \leq l_{j,t} \leq \bar{l}_j & t=1,\ldots,T & j\in J
\end{array}
\tag{3.19}
$$

3.6.4 Nested Benders Decomposition

Um ein lineares MSP zu lösen, ist es vorzuziehen, einen Algorithmus zu benutzen, der die spezielle Struktur des MSPs ausnutzt. Eine Möglichkeit ist die sogenannte nested Benders decomposition, welche das lineare MSP in Teilprobleme zerlegt, die mit den Knoten des Szenariobaumes verbunden sind. Das Originalproblem wird dann gelöst mittels Durchquerung des Szenariobaumes und Lösung der Teilprobleme. Ein Knoten n gibt dabei primale Information an seine Kinder in Form eines Entscheidungsvektors $x^{p(n)}$ weiter, wobei die Kinder duale Informationen an ihre Eltern in der Form von Schnitten weitergeben.

Um die Notation zu vereinfachen, nehmen wir an, daß das lineare MSP die Form

$$
\begin{array}{llll}
\min\limits_{x} & \sum_{t=1}^{T}\mathbb{E}\left[\langle c_t(\zeta_t), x_t\rangle\right] & & \\
\text{s.d.} & \sum_{\tau=1}^{t} A_{t,\tau}(\zeta_t)\, x_\tau \leq b_t(\zeta_t) & \mathbb{P}\text{-f.s.} & t=1,\ldots,T \\
& x_t \in X_t(\zeta_t) & \mathbb{P}\text{-f.s.} & t=1,\ldots,T \\
& x_t = \mathbb{E}\left[x_t \,|\mathcal{F}_t\right] & & t=1,\ldots,T
\end{array}
\tag{3.20}
$$

hat, mit $c_t : \mathbb{R}^{s_t} \to \mathbb{R}$, $A_{t,\tau} : \mathbb{R}^{\sum_{\tau'=1}^{t} s_{\tau'}} \to \mathbb{R}^{d_t\times r_\tau}$, $\tau = 1,\ldots,t$, $b_t : \mathbb{R}^{\sum_{\tau=1}^{t} s_\tau} \to \mathbb{R}^{d_t}$ und $X_t(\zeta_t)$ ist polyhedral für alle $\zeta_t \in \zeta_t(\Omega)$, $t = 1,\ldots,T$. Dann lässt sich die Szenariobaumformulierung schreiben als

$$
\begin{array}{lll}
\min\limits_{x} & \sum_{n\in\bar{N}} \pi_n \left\langle c_{t(n)}(\zeta^n), x^n\right\rangle & \\
\text{s.d.} & \sum_{\tau=1}^{t(n)} A_{t(n),\tau}(\zeta^n)\, x^{n-t(n)+\tau} \leq b_{t(n)}(\zeta^n) & n\in\bar{N} \\
& x^n \in X_{t(n)}(\zeta^n) & n\in\bar{N}
\end{array}
\tag{3.21}
$$

Für einen Knoten $n \in \bar{N}$ und Entscheidungsvektor $x^{p(n_-)}$ für den Pfad von der Wurzel bis zum

Knoten n_-, betrachten wir nun das folgende Teilproblem:

$$\begin{array}{ll} \min\limits_{x^n} & \left\langle c_{t(n)}\left(\zeta^n\right), x^n\right\rangle + \theta^n \\ & Q\left(x^{p(n_-)}, x^n\right) \leq \theta^n \\ \text{s.d.} & A_{t(n),t}\left(\zeta^n\right) x^n \leq b_{t(n)}\left(\zeta^n\right) - \sum\limits_{\tau=1}^{t(n)-1} A_{t(n),\tau}\left(\zeta^n\right) x^{n-t(n)+\tau} \\ & x^n \in X_{t(n)}\left(\zeta^n\right) \end{array} \tag{3.22}$$

wobei

$$\begin{array}{lll} Q\left(x^{p(n)}\right) & := & \min\limits_{\left(x^{n_+}\right)_{n_+\in N_+(n)}} \sum\limits_{n_+\in N_+(n)} \pi_{n_+}\left\langle c_{t(n)+1}\left(\zeta^{n_+}\right), x^{n_+}\right\rangle \\ \text{s.d.} & & A_{t(n_+),t(n_+)}\left(\zeta^{n_+}\right) x^{n_+} \leq b_{t(n_+)}\left(\zeta^{n_+}\right) - \sum\limits_{\tau=1}^{t(n)} A_{t(n_+),\tau}\left(\zeta^{n_+}\right) x^{n-t(n)+\tau} \quad n_+ \in N_+(n) \\ & & x^{n_+} \in X_{t(n_+)}\left(\zeta^{n_+}\right) \quad n_+ \in N_+(n) \end{array} \tag{3.23}$$

Da die Lösung von (3.22) zu teuer ist (eine Auswertung von $Q\left(x^{p(n_-)}, x^n\right)$ erfordert die Lösung von $|N_+(n)|$ Teilproblemen), wird es relaxiert, indem die Restriktion $Q\left(x^{p(n)}\right) \leq \theta^n$ durch eine Menge von linearen Schnitten $c\left(x^n, \theta^n\right) \leq 0$, $c \in \mathcal{C}_n$, ersetzt wird. Die relaxierte Form von (3.22) ist daher

$$\begin{array}{lll} \min\limits_{x^n} & \left\langle c_{t(n)}\left(\zeta^n\right), x^n\right\rangle + \theta^n & \\ & c\left(x^n, \theta^n\right) \leq 0 & c \in \mathcal{C}_n \\ \text{s.d.} & A_{t(n),t}\left(\zeta^n\right) x^n \leq b_{t(n)}\left(\zeta^n\right) - \sum\limits_{\tau=1}^{t(n)-1} A_{t(n),\tau}\left(\zeta^n\right) x^{n-t(n)+\tau} & \\ & x^n \in X_{t(n)}\left(\zeta^n\right) & \end{array} \tag{3.24}$$

Ein Schnitt wird konstruiert durch Lösung des Problemes (3.23), oder seinem dualen Äquivalent, für einen gegebenen Entscheidungsvektor $\hat{x}^{p(n)}$ und unter Ausnutzung der Eigenschaft, daß Q konvex ist. Wenn (3.23) unzulässig ist, so ist sein duales Problem unbeschränkt, und ein Zulässigkeitsschnitt kann abgeleitet werden. Wenn andererseits (3.23) zulässig ist, so erhält man einen Optimalitätsschnitt. Anstatt (3.23) für $n \in \bar{N}$ zu lösen, kann man die Separabilität von (3.23) ausnutzen, und das Problem (3.24) für alle $n_+ \in N_+(n)$ lösen und Optimalitäts- und Zulässigkeitsschnitte aus ihrer Lösung konstruieren. Wir formalisieren diese Gedanken in einem Algorithmus, der für jeden Knoten aufgerufen wird, sobald er aktiviert wird:

Algorithmus 3.12 (Unterprogramm der nested Benders decomposition [20, Kapitel 3, Abschnitt 5.2]). Sei $n \in \bar{N}$.

```
if t(n) > 1 then
   Erhalte eine approximative Lösung x^{p(n_-)} von den Vorgängerknoten.
end if
if t(n) < T then
   Erhalte Optimalitäts- und Zulässigkeitsschnitte von den Nachfolgerknoten N_+(n).
end if
Formuliere und löse das Problem (3.24).
if (3.24) zulässig then
   Speichere einen optimalen Lösungspunkt x^n.
   if t(n) > 1 and der Optimalwert wurde verbessert then
      Konstruiere einen Optimalitätsschnitt.
```

12: **end if**
13: **else** {(3.24) unzulässig}
14: **if** $t(n) > 1$ **then**
15: Konstruiere einen Zulässigkeitsschnitt.
16: **else**
17: **stop**: (3.21) ist unzulässig.
18: **end if**
19: **end if**

Die Reihenfolge, in der Algorithmus 3.12 für die Knoten aufgerufen wird, wird durch das *sequencing protocol* bestimmt. In [4, 5] werden alle Knoten, die die gleiche Zeitstufe t gemeinsam haben, zur gleichen Zeit behandelt. Für solche Knoten, deren Vorgänger ein zulässiges Teilproblem besitzen, kann der Algorithmus 3.12 aufgerufen werden und Optimalitäts- oder Zulässigkeitsschnitte können erzeugt werden. Wenn Teilprobleme (3.24) zulässig waren, so ist es möglich, mit der nächsten Zeitstufe $t+1$ fortzufahren. Anderenfalls muss man zur Zeitstufe $t-1$ zurückkehren und die entsprechenden Teilprobleme dort erneut lösen.

Der Algorithmus kann gestoppt werden, wenn jedes Teilproblem gelöst werden kann ohne einen neuen Schnitt zu erzeugen. Dann wurde auch das komplette Problem (3.21) gelöst. Das andere Haltekriterium ist Zeile 17 von Algorithmus 3.12, welcher die Unzulässigkeit von (3.21) erkennt.

Kapitel 4

Numerische Ergebnisse

4.1 Software ADoSP

Das Softwarepaket ADoSP (Adaptive Discretization of Stochastic Programs) in seiner jetzigen Form ist eine Implementierung des Algorithmusses 3.11 für lineare mehrstufige stochastische Programme, ähnlich zu (MILMSP). ADoSP ist in der Lage, mit Modellen der Form

$$\begin{array}{lll}
\min\limits_{x} & \mathbb{E}\left[\sum\limits_{k=1}^{p} \langle c_k(\xi), x_{I_k}\rangle\right] & \\
& \underline{b}_t(\zeta_t) \leq \sum\limits_{\tau=1}^{t}\sum\limits_{k=1}^{p} A_{t,\tau,k}(\zeta_t)\, x_{I_{\tau,k}} \leq \bar{b}_t(\zeta_t) & \mathbb{P}\text{-f.s.} \quad t=1,\ldots,T \\
\text{s.d.} & \underline{\tilde{b}}_{t,k}(\zeta_t) \leq \sum\limits_{\tau=1}^{t} \tilde{A}_{t,\tau,k}(\zeta_t)\, x_{I_{\tau,k}} \leq \bar{\tilde{b}}_{t,k}(\zeta_t) & \mathbb{P}\text{-f.s.} \quad k=1,\ldots,p \quad t=1,\ldots,T \\
& \underline{x} \leq x \leq \bar{x} & \mathbb{P}\text{-f.s.} \\
& x_{I_{t,k}} = \mathbb{E}\left[x_{I_{t,k}} \,|\mathcal{F}_t\right] & k=1,\ldots,p \quad t=1,\ldots,T
\end{array} \tag{4.1}$$

umzugehen, wobei die Schranken $\underline{b}_t(\cdot)$, $\underline{\tilde{b}}_{t,k}(\cdot)$, $\underline{x}$, $\bar{b}_t(\cdot)$, $\bar{\tilde{b}}_{t,k}(\cdot)$, $\bar{x}$ auch unendliche Werte annehmen können.

Die aktuelle Version von ADoSP besteht aus ca. 6000 Zeilen C++-Code. Modelle können im GAMS-Format ([11]) übergeben werden. Um auftretende LP's zu lösen, benutzt ADoSP die COIN-OSI-Bibliothek ([2]), welche selbst wieder eine Schnittstelle zu CPLEX ([3]) bietet.

4.1.1 Eingabe

Für die Eingabe des Modells in ADoSP wurde eine GAMS-Schnittstelle ([11]) implementiert. Zusätzliche Information über Zeitstufen und Blockstruktur der Variablen muss separat bereitgestellt werden. Die entsprechenden Informationen für die Restriktionen werden automatisch aus den Dünnbesetztheitsinformationen gewonnen.

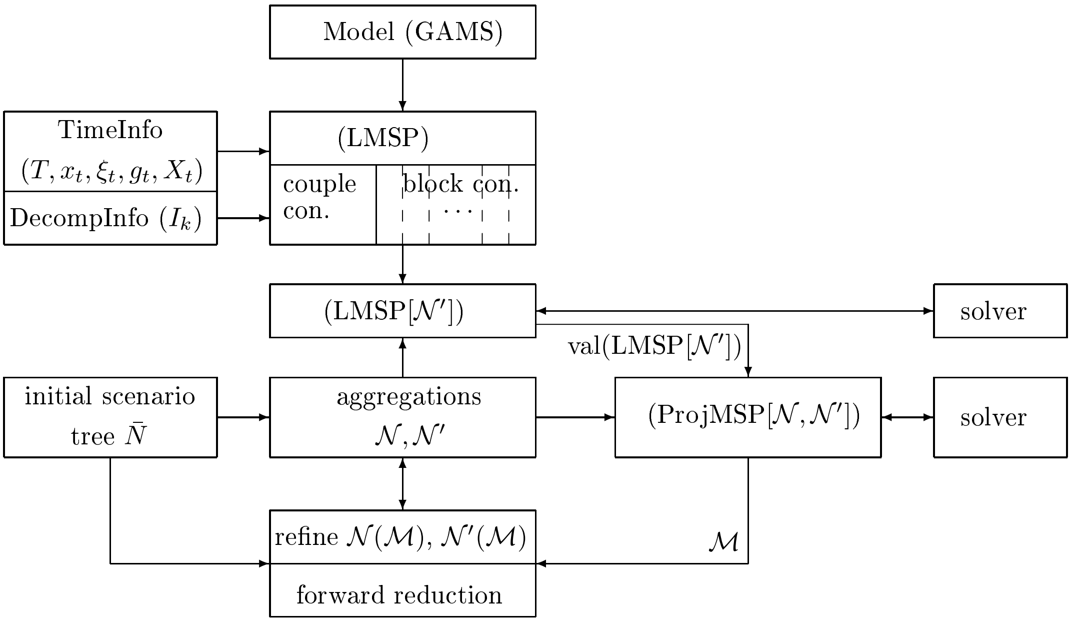


Abbildung 4.1: Design der Software ADoSP

Modell

Das Problem (4.1) wird formuliert als ein NLP in den Variablen x und ξ,

$$\begin{array}{ll} \min\limits_{x,\xi} & f\left(x;\xi\right) \\ \text{s.d.} & \begin{array}{r} \underline{g} \leq g\left(x;\xi\right) \leq \bar{g} \\ \underline{x} \leq x \leq \bar{x} \end{array} \end{array}$$

und die folgenden Annahmen werden gemacht:

- Die Funktion $f\left(x,\xi\right)$ ist gegeben in der Form $y = f\left(x,\xi\right)$, wobei y eine freie Variable ist. Diese Variable wird dann von ADoSP entfernt.
- $\nabla_x f\left(\cdot,\xi\right)$ und $\nabla_x g\left(\cdot,\xi\right)$ sind konstant, d.h. nachdem ξ durch Werte aus einer endlichen Menge von Szenarien ersetzt wurde, erhält man ein lineares Programm.
- Der beschreibende Text für die Entscheidungsvariablen x beginnt mit “desc”, der beschreibende Text für die Datenvariablen beginnt mit “data”. Ein beschreibender Text für y kann weggelassen werden.

Zeitstufen

Eine Datei, welche Informationen über die Zeitstufen enthält, muss separat zur Verfügung gestellt werden.

Die erste (nichtleere) Zeile enthält die Anzahl T der Zeitstufen. Danach werden Informationen über die Variablen, die zu jeder Zeitstufe gehören, angegeben. Eine Zeitstufe wird mit dem Schlüsselwort “NEXT” eingeführt (auch für die erste Zeitstufe). Danach kommen die Namen der Variablen in x_t und ξ_t.

Eine Restriktion wird einer Zeitstufe zugewiesen, indem überprüft wird, welche Variablen in ihr auftreten ($\exists y : \nabla_{y_i} g\left(y\right) \neq 0$), d.h. eine Restriktion wird der Zeitstufe t_* zugewiesen, wenn sie von x_{t_*} oder ξ_{t_*} abhängt, aber nicht von x_t oder ξ_t für $t > t_*$.

Dekomposition

Weiterhin muss eine Datei mit Informationen zur Blockstruktur bereitgestellt werden. Die erste (nichtleere) Zeile enthält die Anzahl p von Blöcken. Danach werden Informationen über die Variablen, die zu jedem Block gehören, angegeben. Ein Block wird mit dem Schlüsselwort "NEXT" eingeleitet (auch für den ersten Block). Danach kommen die Namen der Variablen in x_{I_k}.

Eine Restriktion wird einem Block zugewiesen, indem überprüft wird, welche Variablen in ihr auftreten, d.h. eine Restriktion wird dem Block k_* zugewiesen, wenn sie nur von $x_{I_{k_*}}$ (und ξ) abhängt, aber unabhängig ist von allen x_{I_k}, $k \neq k_*$. Wenn eine Restriktion keinem Block zugewiesen werden kann, so wird sie im folgenden als Kopplungsrestriktion behandelt.

Anfänglicher Szenariobaum

Letztendlich muss der anfängliche Szenariobaum $\bar{N}$ in einer Datei zur Verfügung gestellt werden. Die erste (nichtleere) Zeile enthält die Anzahl $|\bar{N}|$ der Knoten . Darauf folgen Informationen über jeden Knoten. Es wird angenommen, daß die Knoten sequentiell durchnummeriert sind. Jede Zeile besteht aus dem Abstand zum Vorgängerknoten $n - n_-$, der Knotenwahrscheinlichkeit π_n und den Szenariowerten ξ^n. Für den Wurzelknoten $n = 0$ kann der Abstand zum Vorgängerknoten eine beliebige Zahl sein, darf aber nicht ausgelassen werden.

Beispiel

Als Beispiel betrachte man das Modell (3.19) für das Strommanagement unter unsicherer Last bei einem Zeithorizont von einer Woche mit stündlicher Diskretisierung. Das GAMS-Modell ist:

```
SET i Kohlekraftwerke / Bk_Lau_1, Bk_Lau_3, GuD_Kue, GT_Vock / ;
SET j Pumpspeicherwerke / GOLD, MARK / ;
SET t Timestages / 1*168 /;
PARAMETERS b(i) Brennstoffkosten Kohlekraftwerk i / ... / ;
PARAMETERS p_min(i) min. Leistung Kohlekraftwerk i / ... / ;
PARAMETERS p_max(i) max. Leistung Kohlekraftwerk i / ... / ;
PARAMETERS eta(j) Wirkungsgrad Pumpen PSW j / ... / ;
PARAMETERS w_max(j) max. Pumpleistung PSW j / ... / ;
PARAMETERS v_max(j) max. Turbinenleistung PSW j / ... / ;
PARAMETERS l_max(j) max. Fuellstand PSW j / ... / ;
PARAMETERS l_in(j) Fuellstand zu Beginn PSW j; l_in(j) = 0.9 * l_max(j) ;
PARAMETERS l_end(j) Fuellstand am Ende PSW j; l_end(j) = l_in(j) ;
VARIABLES
d(t) data Lastdaten fuer eine Woche
r(t) data Reserveleistung
p(i,t) deci Leistungswerte der thermischen Bloecke i
v(j,t) deci Turbinenwerte der Pumpspeichereinheiten j
w(j,t) deci Pumpwerte der Pumpspeichereinheiten j
l(j,t) deci Fuellstaende der Pumpspeichereinheiten j
cost Gesamte Kosten (Zielfunktionswert) ;
POSITIVE VARIABLES p, v, w, l ;
EQUATIONS
ziel_funktion Zielfunktion
l_anf_res(j) Bedingung Anfangs-Fuellstand
l_kopp_res(j,t) Zeitkopplung der Fuellstaende
last_res(t) Bedingung Lastdeckung
reserve_res(t) Bedingung Reserveforderung ;
```

```
ziel_funktion .. cost =e= sum((i,t),b(i)*p(i,t));
loop(t,
p.lo(i,t)=p_min(i); p.up(i,t)=p_max(i);
v.up(j,t)=v_max(j); w.up(j,t)=w_max(j);
l.up(j,t)=l_max(j);
);
l_anf_res(j) .. l(j,'1') =e= l_in(j) - (v(j,'1')-eta(j)*w(j,'1')) ;
l.fx(j,'168')=l_end(j);
l_kopp_res(j,t)$(ord(t)>1) .. l(j,t) =e= l(j,t-1) - (v(j,t)-eta(j)*w(j,t)) ;
last_res(t) .. sum(i,p(i,t)) + sum(j,v(j,t)-w(j,t)) =g= d(t) ;
reserve_res(t) .. sum(i,p_max(i)-p(i,t)) =g= r(t) ;
MODEL veag_small_stoch_last / all / ;
```

Die Information über die Zeitstufen wird mittels folgender Datei übergeben:

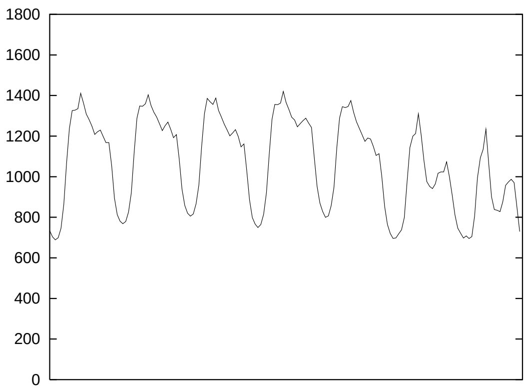
```
168
NEXT
d(1) r(1)
p(Bk_Lau_1,1) p(Bk_Lau_3,1) p(GuD_Kue,1) p(GT_Vock,1)
v(GOLD,1) w(GOLD,1) l(GOLD,1)
v(MARK,1) w(MARK,1) l(MARK,1)
NEXT
d(2) r(2)
p(Bk_Lau_1,2) p(Bk_Lau_3,2) p(GuD_Kue,2) p(GT_Vock,2)
v(GOLD,2) w(GOLD,2) l(GOLD,2)
v(MARK,2) w(MARK,2) l(MARK,2)
NEXT
...
NEXT
d(168) r(168)
p(Bk_Lau_1,168) p(Bk_Lau_3,168) p(GuD_Kue,168) p(GT_Vock,168)
v(GOLD,168) w(GOLD,168) l(GOLD,168)
v(MARK,168) w(MARK,168) l(MARK,168)
```

Die Blockstruktur des Modells ist bestimmt durch die Wärmekraftwerke und Pumpspeicherwerke. Die Datei mit der Dekompositionsinformation hat die Form

```
6
NEXT
p(Bk_Lau_1,1) p(Bk_Lau_1,2) ... p(Bk_Lau_1,168)
NEXT
p(Bk_Lau_3,1) p(Bk_Lau_3,2) ... p(Bk_Lau_3,168)
NEXT
p(GuD_Kue,1) p(GuD_Kue,2) ... p(GuD_Kue,168)
NEXT
p(GT_Vock,1) p(GT_Vock,2) ... p(GT_Vock,168)
NEXT
v(GOLD,1) w(GOLD,1) l(GOLD,1)
v(GOLD,2) w(GOLD,2) l(GOLD,2)
...
v(GOLD,168) w(GOLD,168) l(GOLD,168)
NEXT
v(MARK,1) w(MARK,1) l(MARK,1)
v(MARK,2) w(MARK,2) l(MARK,2)
...
v(MARK,168) w(MARK,168) l(MARK,168)
```

4.2 Strommanagement in einem hydro-thermalen System unter unsicherer Last

ADoSP wurde für das Modell (3.19) gestartet, um einen optimalen Entscheidungsprozess für den Betrieb einer Sammlung von Kohlekraftwerken und Pumpspeicherwerken unter unsicherer Last zu finden, so daß die erwarteten Brennstoffkosten minimiert werden, siehe Abschnitt 3.6.3. Ein Szenario der elektrischen Last für eine Woche ist in der Abbildung rechts dargestellt. Die sieben Tage sind gut unterscheidbar.
Zwei Szenariobäume für die Modellierung der Stochastik in der elektrischen Last für eine Woche waren verfügbar. Der erste Baum besteht aus 64 Szenarien und 3048 Knoten, der zweite aus 729 Szenarien und 22623 Knoten, siehe Abbildungen 4.2 und 4.3 für eine Darstellung.



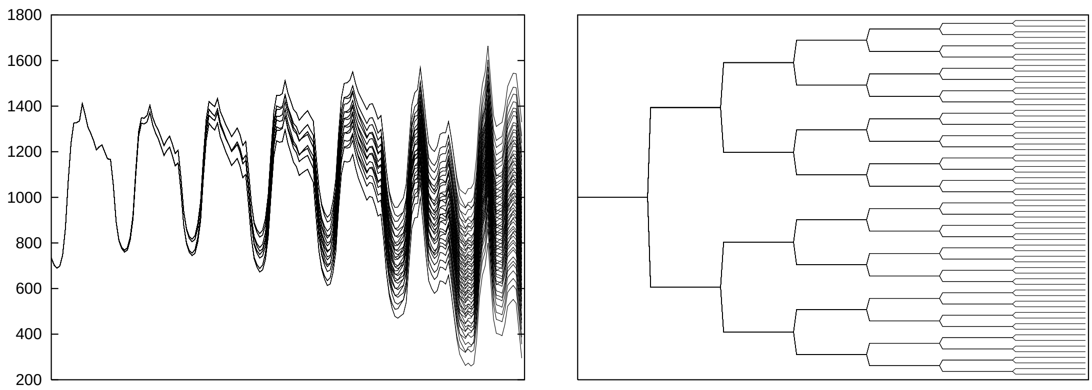


Abbildung 4.2: Links 64 Szenarien der elektrischen Last für eine Woche. Rechts der dazugehörige Szenariobaum.

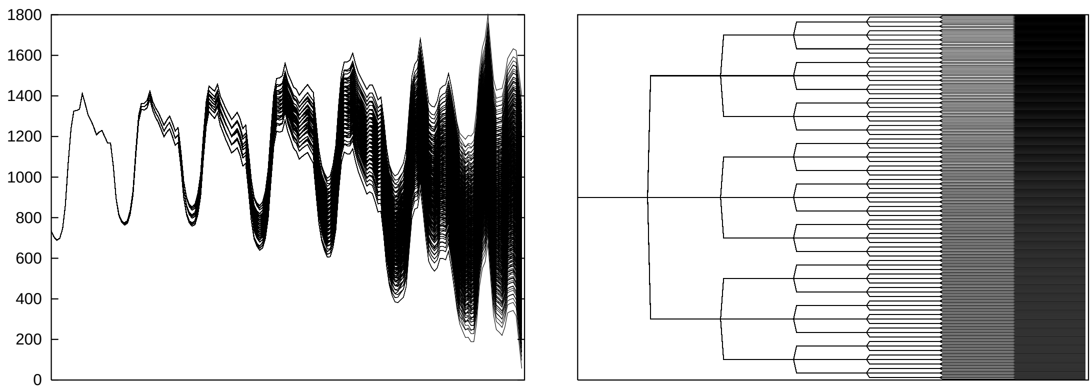


Abbildung 4.3: Links 729 Szenarien der elektrischen Last für eine Woche. Rechts der dazugehörige Szenariobaum.

Wenn ADoSP für den Baum aus 64 Szenarien und Parametern $\sigma_P = 0.001$, $\sigma_D = 0.01$, $\bar{m} = 2$, $s = 5$ gestartet wird, so resultiert dies in einem Baum aus 25 Szenarien, siehe Abbildung 4.4. Abbildung 4.5 zeigt den Verlauf der Optimalwerte $\mathrm{val}\,(\mathrm{LMSP}[\mathcal{N}])$ und $\mathrm{val}\,(\mathrm{LMSP}\,[\mathcal{N}'])$ und wie sich die Größen $\frac{\mathrm{val}(\mathrm{LMSP}[\mathcal{N}])}{\mathrm{val}(\mathrm{LMSP}[\mathcal{N}'])}$ und $\frac{\mathcal{D}(\mathcal{N}')}{\mathcal{D}_0}$ während des Iterationsprozesses entwickeln. Im rechten Bild in Abbildung 4.5 kann man beobachten, in welcher Iteration die Bedingung $\frac{\mathrm{val}(\mathrm{LMSP}[\mathcal{N}'])}{\mathrm{val}(\mathrm{LMSP}[\mathcal{N}])} < 1 + \sigma_P$ fehlgeschlagen ist (wenn die Kurve unter das markierte Intervall fällt, hier die Iterationen 3, 5, 11 und 14). In diesem Fall wählt ADoSP die Menge $\mathcal{M} \subseteq \mathcal{N}$ entsprechend (3.18), so daß $\mathrm{val}\,(\mathrm{LMSP}[\mathcal{N}])$ sich erhöht. Auf der anderen Seite, wenn die Bedingung $\frac{\mathrm{val}(\mathrm{LMSP}[\mathcal{N}])}{\mathrm{val}(\mathrm{LMSP}[\mathcal{N}'])} < 1 + \sigma_P$ verletzt wird (die Kurve überschreitet das markierte Intervall), so löst ADoSP das Problem $(\mathrm{ProjMSP}[\mathcal{N}, \mathcal{N}'])$ und wählt die Menge $\mathcal{M}$ entsprechend Formel (2.2), um eine Harmonisierung von $\mathrm{val}\,(\mathrm{LMSP}[\mathcal{N}])$ und $\mathrm{val}\,(\mathrm{LMSP}\,[\mathcal{N}'])$ zu erreichen. Wenn andererseits $\frac{1}{1+\sigma_P} \leq \frac{\mathrm{val}(\mathrm{LMSP}[\mathcal{N}'])}{\mathrm{val}(\mathrm{LMSP}[\mathcal{N}])} \leq 1+\sigma_P$, aber $\frac{\mathcal{D}(\mathcal{N}')}{\mathcal{D}_0} > \sigma_D$, so wird $\mathcal{N}'$ verfeinert, d.h. die Menge $\mathcal{M} \subseteq \mathcal{N}$ wird so gewählt, daß der Abstand von $\bar{N}\,|_{\mathcal{N}'(\mathcal{M})}$ zu $\bar{N}$ verkleinert wird. In der linken Abbildung kann man diese Iterationen daran erkennen, daß keine Änderung von $\mathrm{val}\,(\mathrm{LMSP}[\mathcal{N}])$ in diesem Fall auftritt.

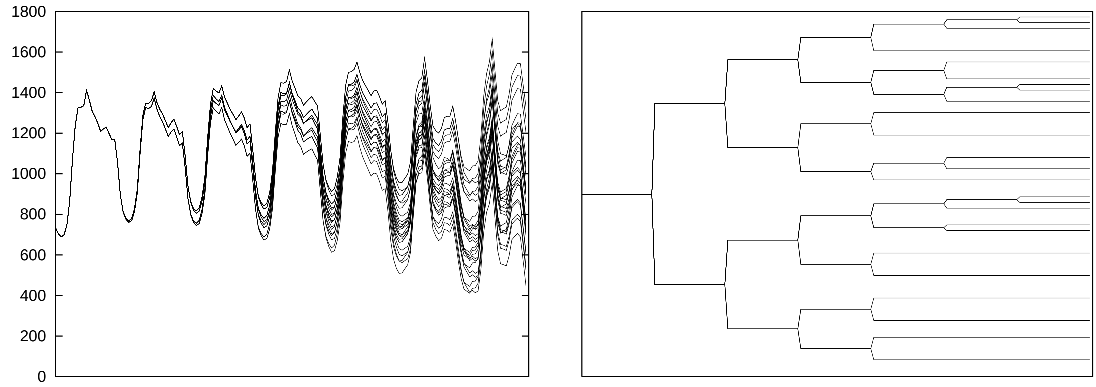


Abbildung 4.4: Reduzierter Szenariobaum aus 25 Szenarien. Der anfängliche Szenariobaum hatte 64 Szenarien. Die Parameter waren $\sigma_p = 0.001$ und $\sigma_D = 0.01$.

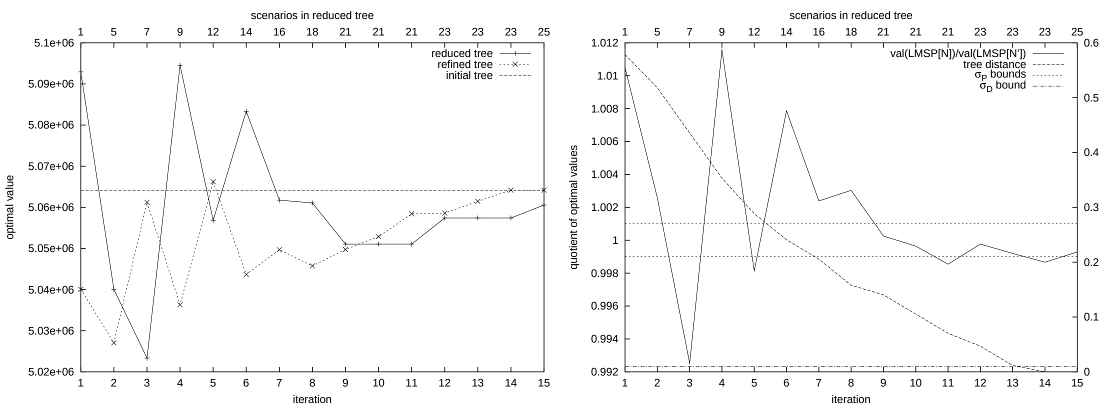


Abbildung 4.5: Iterationsverlauf für einen anfänglichen Szenariobaum aus 64 Szenarien und $\sigma_p = 0.001$, $\sigma_D = 0.01$. Das linke Bild zeigt den Optimalwert bzgl. dem reduzierten Baum $\bar{N}\,|_{\mathcal{N}}$, dem verfeinerten Baum $\bar{N}\,|_{\mathcal{N}'}$ und dem anfänglichen Baum $\bar{N}$ (durchgezogene Linie). Das rechte Bild zeigt den Quotienten $\frac{\mathrm{val}(\mathrm{LMSP}[\mathcal{N}])}{\mathrm{val}(\mathrm{LMSP}[\mathcal{N}'])}$ und markiert das Intervall $\left[\frac{1}{1+\sigma_P}, 1+\sigma_P\right]$ bzgl. der linken Ordinate und zeigt den Quotienten $\frac{\mathcal{D}(\mathcal{N}')}{\mathcal{D}_0}$ und markiert die Schranke σ_D auf der rechten Ordinate.

Um ein Bild des Einflusses der Parameter σ_P und σ_D auf die Anzahl der Szenarien und Knoten im reduzierten Baum zu bekommen, wurden mehrere Läufe mit $\sigma_P \in \{0.1, 0.01, 0.005, 0.001, 0.0005, 0.0001\}$ und $\sigma_D \in \{0.5, 0.1, 0.05, 0.01\}$ durchgeführt. Abbildung 4.9 am Ende dieses Abschnittes zeigt den Iterationsverlauf für eine ausgewählte Menge von Kombinationen der gewählten Parameter, die Tabellen 4.1 fassen die Zahl der Szenarien bzw. Knoten im finalen reduzierten Baum zusammen. Während σ_P offensichtlich einen großen Einfluss auf die endgültige Zahl von Szenarien hat, scheint der Abstand $\mathcal{D}(\mathcal{N}')$, kontrolliert von σ_D, geringere Bedeutung zu haben.

$\sigma_p \backslash \sigma_D$	0.5	0.1	0.05	0.01
0.1	1	1	1	1
0.01	5	5	5	5
0.005	6	6	6	6
0.001	21	23	23	25
0.0005	21	23	23	27
0.0001	39	39	39	41

$\sigma_p \backslash \sigma_D$	0.5	0.1	0.05	0.01
0.1	168	168	168	168
0.01	624	624	624	624
0.005	720	720	720	720
0.001	1728	1824	1824	1872
0.0005	1728	1824	1824	1920
0.0001	2424	2424	2424	2472

Tabelle 4.1: Die Anzahl der Szenarien (links) bzw. Knoten (rechts) des endgültigen reduzierten Baumes $\bar{N}\,|_{\mathcal{N}}$ für $\sigma_P \in \{0.1, 0.01, 0.005, 0.001, 0.0005, 0.0001\}$ und $\sigma_D \in \{0.5, 0.1, 0.05, 0.01\}$.

Wir gehen über zur Reduktion des Baumes aus 729 Szenarien. Wegen der hohen Zahl von Szenarien haben wir auch die Parameter $\bar{m}$ und s erhöht, so daß auch die Zahl der Szenarien in $\bar{N}\,|_{\mathcal{N}}$ und $\bar{N}\,|_{\mathcal{N}'}$ während der Verfeinerungsoperationen schneller ansteigen kann. Für $\sigma_P = 0.001$, $\sigma_D = 0.01$, $\bar{m} = 10$, $s = 25$, endet ADoSP mit einem Baum aus 77 Szenarien, siehe Abbildung 4.6. Abbildung 4.7 zeigt den Verlauf der Optimalwerte $\mathrm{val}\,(\mathrm{LMSP}[\mathcal{N}])$ und $\mathrm{val}\,(\mathrm{LMSP}\,[\mathcal{N}'])$ und wie sich die Größen $\frac{\mathrm{val}(\mathrm{LMSP}[\mathcal{N}])}{\mathrm{val}(\mathrm{LMSP}[\mathcal{N}'])}$ und $\frac{\mathcal{D}(\mathcal{N}')}{\mathcal{D}_0}$ während des Iterationsprozesses entwickelt haben. Man erkennt, daß nach der sechsten Iteration die restlichen Iterationen dazu genutzt wurden, um die Größe $\frac{\mathcal{D}(\mathcal{N}')}{\mathcal{D}_0}$ unter die Schranke σ_D zu drücken. Wenn man den Parameter σ_D auf 0.1 erhöht, so erhält man den gleichen Baum, aber die Iterationszahl sinkt von 22 auf 15, siehe Abbildung 4.8. Schon ein Wert σ_D von 0.3 würde ausreichen, um das gleiche Resultat zu erhalten.

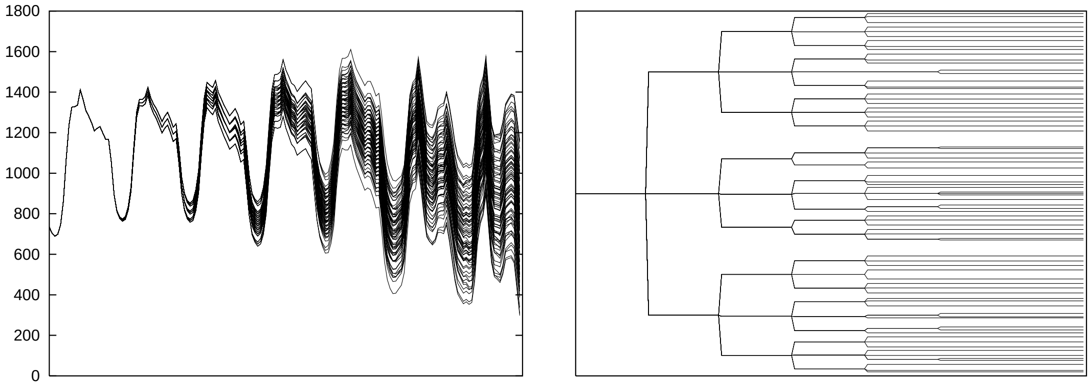


Abbildung 4.6: Reduzierter Szenariobaum aus 77 Szenarien. Der anfängliche Szenariobaum hatte 729 Szenarien. Die Parameter waren $\sigma_p = 0.001$ und $\sigma_D = 0.01$.

Wiederum untersuchen wir den Einfluss der Parameter σ_P und σ_D auf die Anzahl der Szenarien und Knoten im reduzierten Baum. Verschiedene Läufe mit $\sigma_P \in \{0.01, 0.001, 0.0001\}$ und $\sigma_D \in \{0.1, 0.01\}$ wurden durchgeführt. Abbildung 4.10 zeigt den Iterationsverlauf, die Tabellen 4.2 fassen die Anzahl der Szenarien bzw. Knoten im endgültigen reduzierten Baum zusammen.

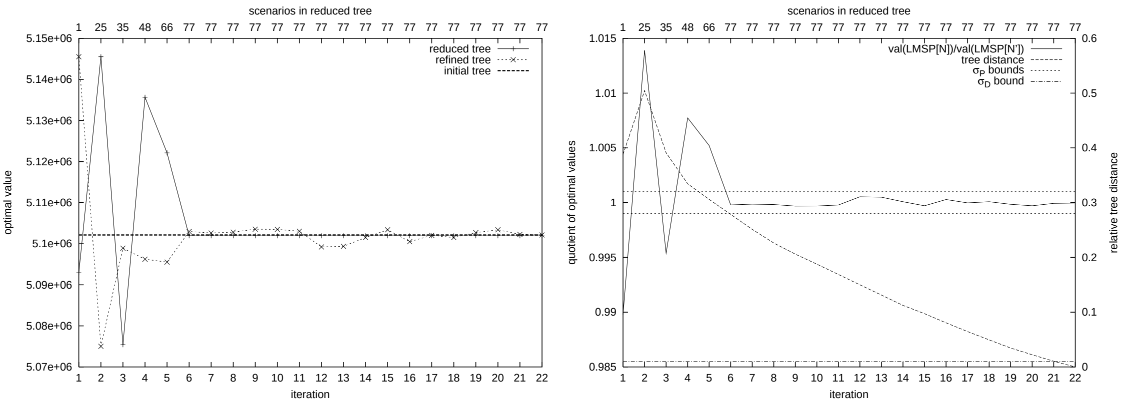


Abbildung 4.7: Iterationsverlauf für einen anfänglichen Szenariobaum aus 729 Szenarien und $\sigma_p = 0.001$, $\sigma_D = 0.01$. Das linke Bild zeigt die Optimalwerte bzgl. dem reduzierten Baum $\bar{N}|_{\mathcal{N}}$, dem verfeinerten Baum $\bar{N}|_{\mathcal{N}'}$ und dem anfänglichen Baum $\bar{N}$ (durchgezogene Linie). Das rechte Bild zeigt die Quotienten $\frac{\text{val}(\text{LMSP}[\mathcal{N}])}{\text{val}(\text{LMSP}[\mathcal{N}'])}$ und $\frac{\mathcal{D}(\mathcal{N}')}{\mathcal{D}_0}$.

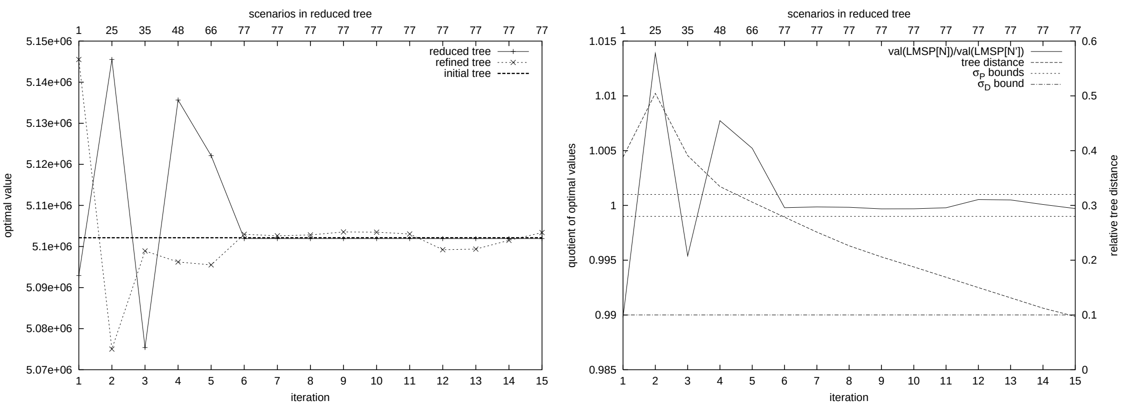


Abbildung 4.8: Iterationsverlauf für einen anfänglichen Szenariobaum aus 729 Szenarien und $\sigma_p = 0.001$, $\sigma_D = 0.1$. Das linke Bild zeigt die Optimalwerte bzgl. dem reduzierten Baum $\bar{N}|_{\mathcal{N}}$, dem verfeinerten Baum $\bar{N}|_{\mathcal{N}'}$ und dem anfänglichen Baum $\bar{N}$ (durchgezogene Linie). Das rechte Bild zeigt die Quotienten $\frac{\text{val}(\text{LMSP}[\mathcal{N}])}{\text{val}(\text{LMSP}[\mathcal{N}'])}$ und $\frac{\mathcal{D}(\mathcal{N}')}{\mathcal{D}_0}$.

$\sigma_p \backslash \sigma_D$	0.1	0.01
0.01	47	47
0.001	77	77
0.0001	294	630

$\sigma_p \backslash \sigma_D$	0.1	0.01
0.01	4152	4152
0.001	6240	6240
0.0001	15672	23856

Tabelle 4.2: Anzahl der Szenarien (links) bzw. Knoten (rechts) des endgültigen reduzierten Szenariobaumes $\bar{N}|_{\mathcal{N}}$ für $\sigma_P \in \{0.01, 0.001, 0.0001\}$ und $\sigma_D \in \{0.1, 0.01\}$.

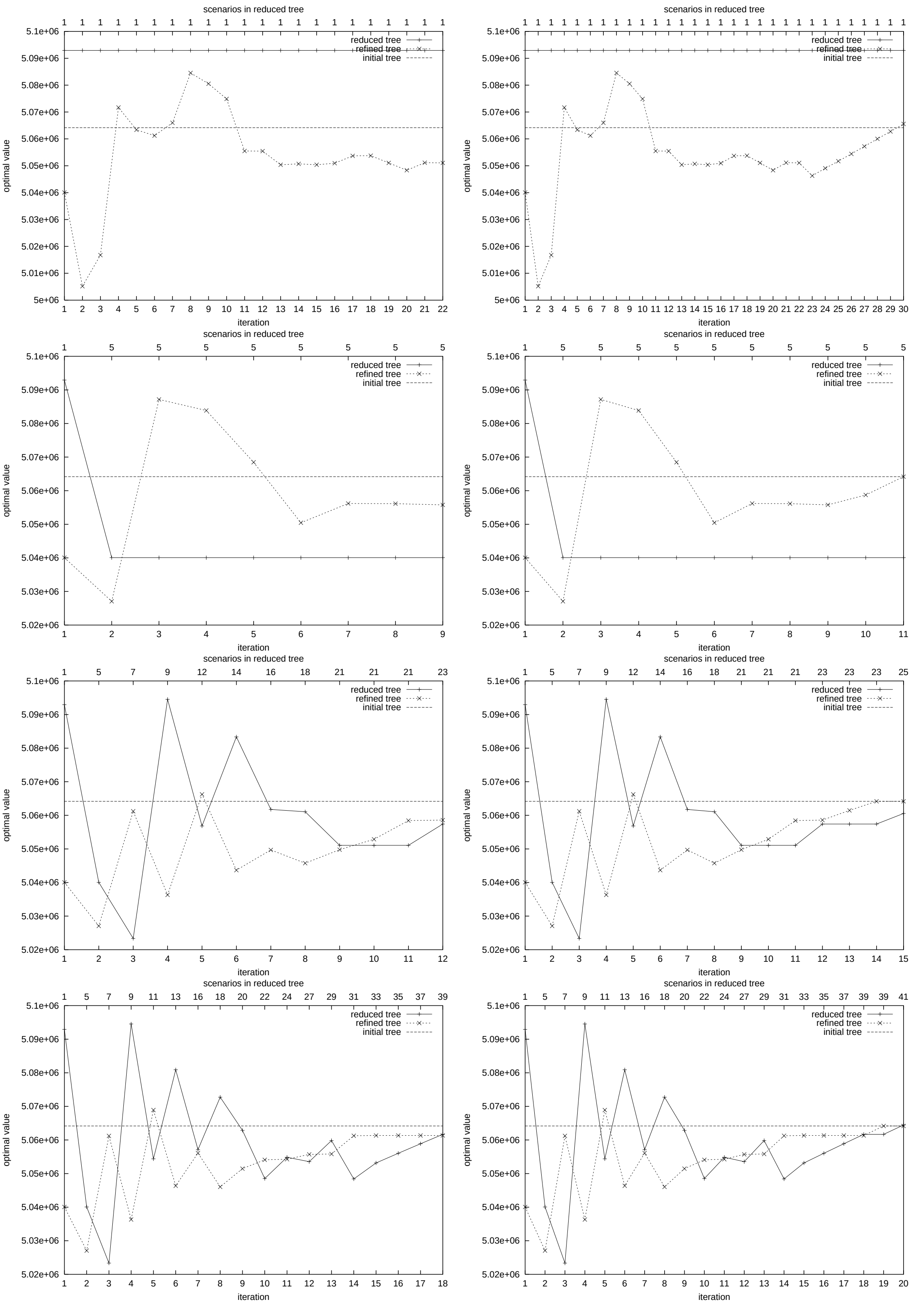


Abbildung 4.9: Iterationsverlauf für $\sigma_P \in \{0.1, 0.01, 0.001, 0.0001\}$ (zeilenweise) und $\sigma_D \in \{0.1, 0.01\}$ (spaltenweise). Der anfängliche Szenariobaum hatte 64 Szenarien.

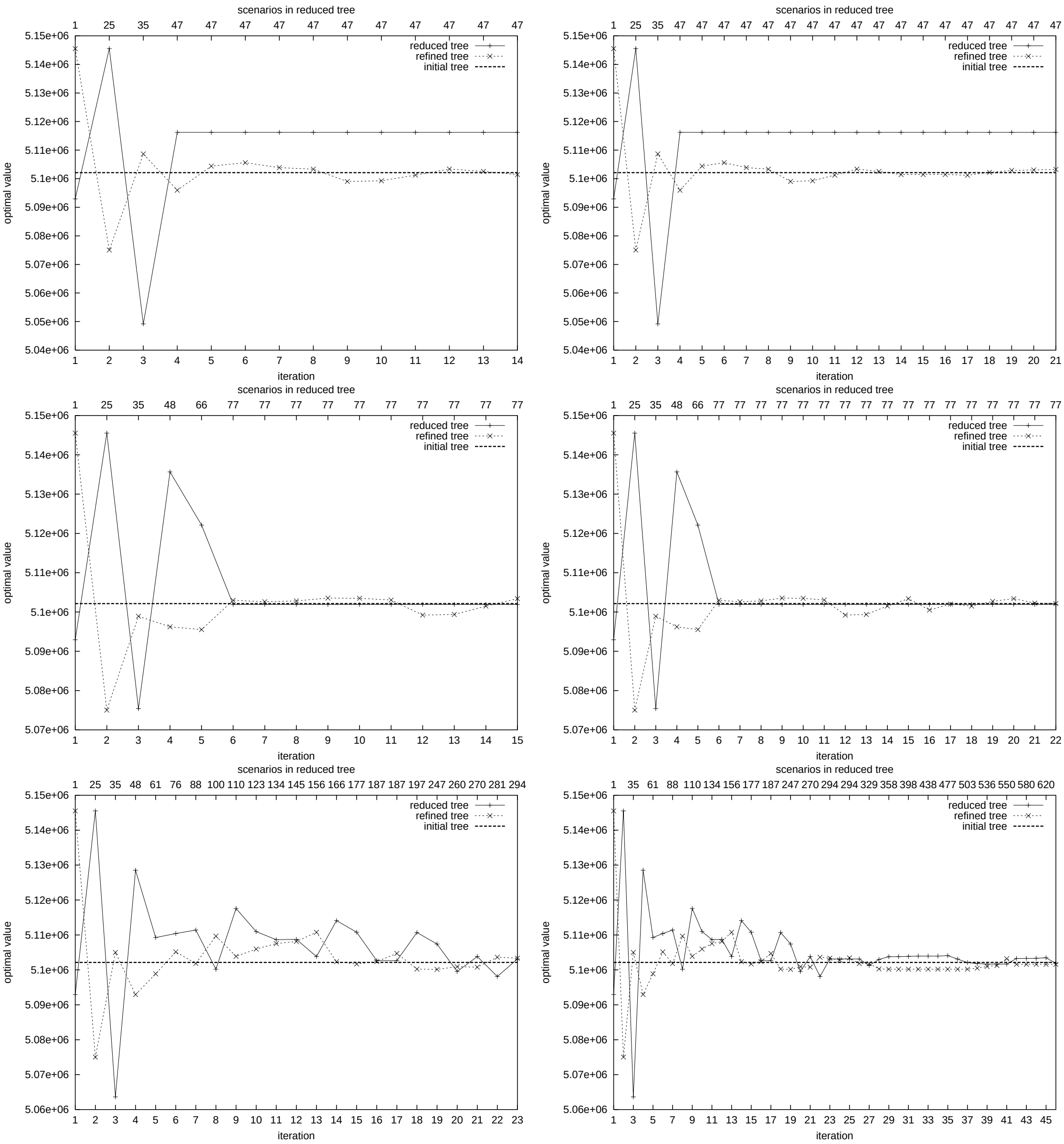


Abbildung 4.10: Iterationsverlauf für $\sigma_P \in \{0.01, 0.001, 0.0001\}$ (zeilenweise) und $\sigma_D \in \{0.1, 0.01\}$ (spaltenweise) . Der anfängliche Szenariobaum hatte 729 Szenarien.

Kapitel 5

Ende

Natürlich haben der präsentierte Algorithmus und seine Implementierung im aktuellen Stadium nicht wenige Schwachpunkte. Wir diskutieren hier einige der aktuellen Nachteile und geben Vorschläge für zukünftige Verbesserungen an.

5.1 Weiterentwicklungen

Anfänglicher Szenariobaum $\bar{N}$

Wie man schon für den Baum aus 729 Szenarien beobachten konnte, ist die Zeit, um Szenarien mittels fast forward selection auszuwählen, ein wichtiger Faktor in der letztendlichen Lösungszeit, wenn die Zahl s von Szenarien, die aus der ursprünglichen Menge N_T ausgewählt werden müssen, sich erhöht. Aus [15] wissen wir, daß fast-forward selection die Zeit $O\left(s^3 + s\,|N_T|^2\right)$ benötigt. Da wir nur eine kleine Zahl von Szenarien in jeder Iteration auswählen, bestünde eine Möglichkeit darin, mit einem relativ kleinen Baum $\bar{N}$ zu beginnen, und dessen Größe während des Iterationsprozesses zu erhöhen. Für eine sehr große Menge $\mathfrak{N}$ von Knoten, die nicht einmal explizit bekannt sein muss, könnte man die Knotenmenge $\bar{N}$ als eine (endliche) Aggregierung von $\mathfrak{N}$ definieren. Während der Verfeinerung von $\mathcal{N}'$ könnte man dann $\bar{N}$ vergrößern, indem die Knoten in den $N' \in \mathcal{N}'$ verfeinert werden, für die $|N'|$ klein ist.

Weiterhin erlaubt dieser Ansatz eine Verallgemeinerung auf stetige Verteilungen, so daß die Konstruktion von Szenarien, oder die Generierung von Szenariobäumen, in den Lösungsprozess integriert wird.

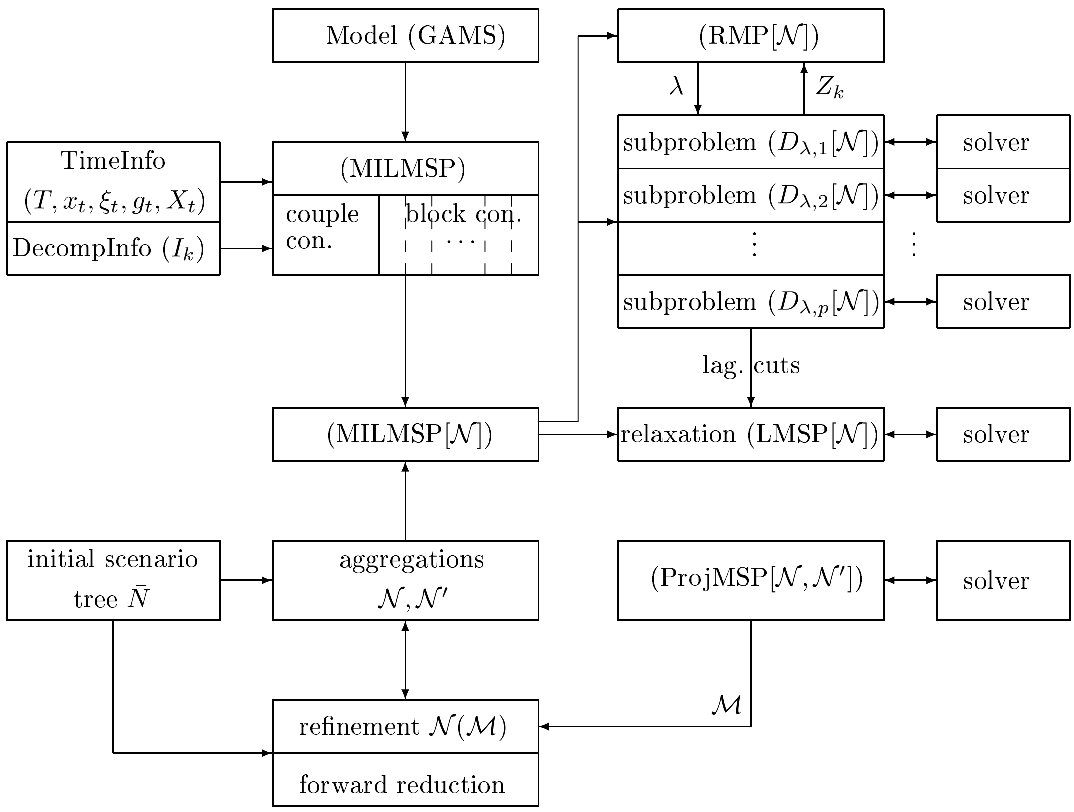
(Proj$[\mathcal{N},\mathcal{N}']$)

Einen noch größeren Einfluss auf die Lösungszeit hat der Aufwand, der in die Lösung des Problems (Proj$[\mathcal{N},\mathcal{N}']$) (bzw. (ProjMSP$[\mathcal{N},\mathcal{N}']$)) gesteckt wird. Da wir nur daran interessiert sind, Mengen $N \in \mathcal{N}$ zu finden, welche wertvoll für eine Verfeinerung sind, ist es möglich, (Proj$[\mathcal{N},\mathcal{N}']$) mit kleinerer Genauigkeit zu lösen. Ein anderer Nachteil von (Proj$[\mathcal{N},\mathcal{N}']$) ist, daß aufgrund der Zielfunktion $\sum_{N\in\mathcal{N}} \|W_N x_N\|_\infty$ eine hohe Interaktion zwischen den Variablen eingeführt wird, die auch noch durch die Restriktion $F_{\mathcal{N}'}(x) - \operatorname{val}(C[\mathcal{N}']) \le \sigma_P\,|\operatorname{val}(C[\mathcal{N}'])|$ verstärkt wird. Im Falle von mehrstufigen stochastischen Programmen verbietet dies die Vererbung der Struktur von (MSP$[\mathcal{N}']$) nach (ProjMSP$[\mathcal{N},\mathcal{N}']$). Daher wäre ein etwas klügeres Vorgehen, vielleicht eine heuristische Methode, zur Auswahl der Menge $\mathcal{M}$ im Fall $\operatorname{val}(C[\mathcal{N}]) - \operatorname{val}(C[\mathcal{N}']) > \sigma_P\,|\operatorname{val}(C[\mathcal{N}'])|$ wünschenswert.

Implementierung der Spaltengenerierung

Das Softwarepaket ADoSP soll für gemischt-ganzzahlige lineare mehrstufige stochastische Programme erweitert werden. Dafür müssen konvexe Relaxierungen (wie (LMSP[$\mathcal{N}$])) eingeführt werden und Schnittgenerierung muss implementiert werden. Die Spaltengenerierung, wie in Abschnitt 2.4.2 vorgeschlagen, ist dafür ein bevorzugter Kandidat. Einige der besten heutigen Löser für gemischt-ganzzahlige lineare Programme sind erfolgreiche Implementierungen der Spaltengenerierungsmethode. Für eine Implementierung dieses Verfahrens müssen viele Aspekte wie die korrekte Initialisierung des (RMP) beachtet werden. Eine andere wichtige Komponente ist die schnelle Lösung der partiellen Lagrangeprobleme $(D_{\lambda,k})$, welche nach einer Aktualisierung der dualen Parameter auftreten. Das Konzept der "erkundeten" Teilprobleme kann genutzt werden, um deren Anzahl zu reduzieren, aber dies ist nur möglich, wenn (RMP) zulässig ist. Darüber hinaus können spezialisierte Löser, welche die Struktur der Teilprobleme effizient ausnutzen, zu einer starken Beschleunigung des kompletten Verfahrens führen.

Ein Design von ADoSP, welches auch Spaltengenerierung beinhaltet, könnte folgende Form haben:



Lagrange-Heuristik

Etwas, was in dieser Arbeit nicht behandelt wurde, ist die Tatsache, daß eine Lösung von $(C[\mathcal{N}])$ für den reduzierten Baum $\mathcal{N}$ in seiner endgültigen Form einen Lösungspunkt liefert, der sich möglicherweise bereits nahe an einem Lösungspunkt von $(P[\mathcal{N}])$ befindet, abhängig von der Qualität der konvexen Relaxierung. Im Falle gemischt-ganzzahliger linearer MSPs erfüllt der Punkt im allgemeinen nicht die Ganzzahligkeitsbedingungen. Daher könnte eine geschickte Rundungsheuristik genutzt werden um einen zulässigen und nahezu optimalen Punkt von (MSP[$\mathcal{N}$]) zu finden.

Wenn Spaltengenerierung benutzt wird, so erfüllen die Punkte $Z_{k,\mathcal{N}}$ bereits die Block-Restriktionen $X_{k,\mathcal{N}}$, $k = 1, \ldots, p$. Daher wäre eine Heuristik, die Startpunkte für eine lokale Suche in $(P[\mathcal{N}])$

liefert, indem sie Extrempunkte so kombiniert, daß die Verletzung der koppelnden Restriktionen klein ist, eine weitere Möglichkeit einen Lösungspunkt von $(P\,[\mathcal{N}])$ zu erhalten. Nowak hat in [17] eine einfache Lagrange-Heuristik vorgeschlagen, die die Extrempunkte aus der Spaltengenerierungsmethode benutzt.

In [12] wird eine Lagrange-Heuristik eingesetzt, um einen primalen Lösungspunkt, den man aus einem proximal-bundle-Verfahren gewonnen hat, zu einem nahezu optimalen zulässigen Punkt von (MSP$[\mathcal{N}]$) umzuformen. Diese Lagrange-Heuristik nutzt die Struktur des Modells aus, um zu entscheiden, welche Binärvariablen umgeschaltet werden sollten. Daher hat sie den Nachteil, daß Änderungen am Modell auch eine Änderung der Heuristik benötigen, liefert auf der anderen Seite aber möglicherweise Lösungskandidaten in viel kürzerer Zeit als eine allgemeine Heuristik.

Nichtlinearität

Letztendlich ist es auch vorstellbar, ADoSP auf den Fall zu erweitern, daß die Menge X_k nicht polyhedral ist, sondern durch nichtlineare Funktionen und Ganzzahligkeitsbedingungen beschrieben wird. Die Berechnung einer linearen äußeren Approximation von X_k liefert ein lineares Problem $(C\,[\mathcal{N}])$, welches weiterhin effizient in großen Dimensionen gelöst werden kann. Wenn die Relaxierung mittels (linearer) Lagrange-Schnitte aktualisiert werden soll, so müssen die nichtlinearen partiellen Lagrangeprobleme $(D_{\lambda,k}\,[\mathcal{N}])$ (global) optimal gelöst werden. Wiederum wäre dies effizient möglich, wenn spezialisierte Lösungsalgorithmen zur Verfügung stehen. Wenn dies nicht der Fall ist, und die Menge X_k nicht konvex ist, müssen allgemeine MINLP-Löser eingesetzt werden, was zur Zeit nur die Behandlung von Problemen kleinerer Größenordnung erlaubt.

5.2 Thesen

- Die Reduktion von Szenariobäumen ist ein wichtiger Faktor bei der effizienten Lösung von stochastischen Programmen.
- Besonders adaptive Diskretisierungsalgorithmen, welche die Erstellung oder Reduktion von Szenariobäumen in den Lösungsprozess eines mehrstufigen stochastischen Programmes integrieren, sind von großem Interesse.
- Der adaptive Diskretisierungsalgorithmus, der in dieser Arbeit vorgeschlagen wurde, liefert eine Erweiterung der backward-reduction und forward-selection Algorithmen aus [12, 15] in dem Sinne, daß eine bessere Abschätzung der Lücke zwischen dem reduzierten und dem anfänglichen Problem möglich ist.
- Um eine Abschätzung für die Abweichung des Optimalwertes vom anfänglichen Problem zu bestimmen, werden konvexe Relaxierungen eingesetzt. Schnittgenerierung spielt eine wichtige Rolle, um die dabei eingeführte Lücke zu reduzieren.
- Spaltengenerierung, basierend auf Lagrange-Relaxierung und -Dekomposition, bildet einen vielversprechenden Ansatz für die Handhabung von stochastischen Programmen hoher Größenordnung.
- Für lineare mehrstufige stochastische Programme wurde ein adaptiver Diskretisierungsalgorithmus erfolgreich implementiert.
- Die adaptive Diskretisierung von Szenariobäumen für stochastische Programme, die in Strommanagementumgebungen auftreten, erlaubt eine Reduktion des Szenariobaumes auf bis zu 10 bis 30% (in der Anzahl der Szenarien), wobei man nur höchstens 0.1% der Genauigkeit im Optimalwert verliert.

Literaturverzeichnis

[1] M. S. Casey, S. Sen, The Scenario Generation Algorithm for Multistage Stochastic Linear Programming, to appear in Mathematics of Operations Research

[2] Computational Infrastructure for Operations Research, `http://www.coin-or.org`

[3] CPLEX, `http://www.ilog.com`

[4] M.A.H. Dempster, R.T. Thompson, EVPI-based importance sampling solution procedures for multistage stochastic programmes on parallel MIMD architectures, Annals of Operations Research, 90 (1999), 161-184

[5] M.A.H. Dempster, R.T. Thompson, Parallelization and Aggregation of Nested Benders Decomposition, Annals of Operations Research, 81, proceedings of the APMOD95 Conference, Brunel University

[6] D. Dentcheva, W. Römisch: Duality gaps in nonconvex stochastic optimization, Mathematical Programming Ser. A 101 (2004), 515-535

[7] J. Dupačová, N. Gröwe-Kuska, W. Römisch, Scenario Reduction in Stochastic Programming - An Approach using Probability Metrics, Mathematical Programming, Ser. A 95 (2003), 493-511

[8] N. C. P. Edirisinghe, Bound-based approximations in multistage stochastic programming: Nonanticipativity aggregation, Annals of Operations Research 85, 103-127 (1999)

[9] A. Eichhorn, N. Gröwe-Kuska, A. Liebscher, W. Römisch, G- Spangardt, I. Wegner, Mean-risk optimization of electricity portfolios, PAMM Proc. Appl. Math. Mech. 4 (2004), 3–6.

[10] I.V. Evstigneev, Measurable selections and dynamic programming, Math. of Operations Research, 1, 1976

[11] General Algebraic Modeling System, `http://www.gams.com`

[12] N. Gröwe-Kuska, H. Heitsch, W. Römisch, Scenario Reduction and Scenario Tree Construction for Power Management Problems, IEEE Bologna Power Tech Proceedings (A. Borghetti, C.A. Nucci, M. Paolone eds.), 2003 IEEE.

[13] N. Gröwe-Kuska, K. C. Kiwiel, M. P. Nowak, W. Römisch, I. Wegner, Power Management in a hydro-thermal system under uncertainty by Lagrangian Relaxation, in: Decision Making under Uncertainty: Energy and Power (C. Greengard, A. Ruszczynski eds.), IMA Volumes in Mathematics and its Applications Vol. 128, Springer-Verlag, New York 2002, 39-70.

[14] N. Gröwe-Kuska, A. Liebscher, M. Lucht, W. Römisch, G. Spangardt, I. Wegner, Mittelfristige risikoorientierte Optimierung von Strombeschaffungs-Portfolios kleinerer Marktteilnehmer, Preprint 03-11, Institut für Mathematik, Humboldt-Universität zu Berlin, 2003

[15] H. Heitsch, W. Römisch, Scenario Reduction Algorithms in Stochastic Programming, Computational Optimization and Applications 24 (2003), 187-206

[16] J. L. Higle, B. Rayco, S. Sen, Stochastic Scenario Decomposition for Multi-Stage Stochastic Programs, submitted to Annals of Operations Research

[17] I. Nowak, Relaxation and Decomposition Methods for Mixed Integer Nonlinear Programming, Habilitationsarbeit, Humboldt-Universität zu Berlin, 2004

[18] S. T. Rachev, Probability metrics and the stability of stochastic models, Wiley, 1991

[19] W. Römisch, Vorlesung "Stochastische Optimierung", Humboldt-Universität zu Berlin, Wintersemester 2002/03

[20] A. Ruszczyński, A. Shapiro, Stochastic Programming, Handbooks in Operations Research and Management Science, Volume 10, Elsevier Science, 2003