

10 Ein Dekompositionsverfahren für stochastische Optimierungsprobleme mit rekombinierenden Szenariobäumen

Küchler, C. ; Vigerske, S.

Kurzfassung: Dieses Kapitel stellt ein Dekompositionsverfahren für lineare mehrstufige stochastische Programme vor, welches auf rekombinierenden Szenariobäumen beruht. Rekombinierende Szenariobäume sind numerisch vorteilhaft, da sie es erlauben, ein exponentielles Wachstum der Knotenzahl des Optimierungsproblems mit der Anzahl der Zeitstufen zu verhindern. Es wird gezeigt, wie sich mittels eines geeigneten Dekompositionsansatzes dieser Vorteil auch unter Gegenwart zeitkoppelter Restriktionen bewahren lässt. Das vorgestellte Verfahren ist verwandt mit der Nested Benders Dekomposition und nutzt die spezielle Struktur rekombinierender Szenariobäume zur simultanen Generierung von Schnittebenen und zum "dynamischen Rekombinieren" von Szenarien. Weiter wird ein Verfahren zur Erstellung von rekombinierenden Bäumen vorgestellt.

Schlagworte: *Mehrstufige Stochastische Optimierung, Nested Benders Dekomposition, Rekombinierende Szenariobäume*

10.1 Einleitung

In vielen Anwendungen ist es notwendig, Entscheidungen zu treffen, deren Konsequenzen von unbekanntem, erst in der Zukunft beobachtbaren Größen abhängen. Die Suche nach einer optimalen Entscheidung kann als ein Optimierungsproblem betrachtet werden, das Unsicherheit hinsichtlich eines oder mehrerer Modellparameter beinhaltet. Oft sind gewisse statistische Informationen über diese "stochastischen" Parameter verfügbar, beispielsweise aus historischen oder simulierten Daten gewonnene empirische Wahrscheinlichkeitsverteilungen. Dann ist es möglich, diese Parameter durch Zufallsvariablen oder stochastische Prozesse zu ersetzen. Auf diese Weise erhält man ein stochastisches Optimierungsproblem [19], dessen Lösungen Unterstützung bei der konkreten Entscheidungsfindung bieten können. Ein Beispiel hierfür ist das in Kapitel 8 betrachtete Modell, bei dem die eingespeiste Windenergie, die Strompreise und die Stromnachfrage als stochastische Größen gegeben sind.

Zwei- und mehrstufige stochastische Optimierungsprobleme zeichnen sich dadurch aus, dass die in der ersten (Zeit-)Stufe zu treffenden Entscheidungen ohne Kenntnis der Realisierung der stochastischen Größen zu treffen sind, während die in der zweiten Stufe zu treffenden Entscheidungen von den bis zu diesem Zeitpunkt beobachteten stochastischen Größen abhängen dürfen. Mehrstufige Probleme beinhalten weitere Zeitstufen mit erst dann beobachtbaren zufälligen Größen. Ihre Lösungen bestehen aus einer Abfolge von Entscheidungen, die jeweils wieder nur von den bis zum jeweiligen Entscheidungszeitpunkt beobachtbaren Größen abhängen dürfen.

Mehrstufige stochastische Optimierungsprobleme können im Allgemeinen nur durch numerische Methoden gelöst werden. Dazu wird der zugrunde liegende stochastische Prozess, welcher beispielsweise durch ein Zeitreihenmodell gegeben sein kann, durch einen Prozess approximiert, der nur endlich viele Werte annimmt. Letzterer kann durch einen Szenariobaum repräsentiert werden (vgl. Kapitel 11). Da die Anzahl der Szenarien in einem solchen Baum exponentiell mit der Länge des betrachteten Zeithorizonts wachsen kann, vgl. Abbildung 10.1, werden zumeist nur Modelle mit einer geringen Anzahl von Zeitstufen oder einer deutlich reduzierten Anzahl von Szenarien betrachtet.

Eine Reduktion der Zeitstufen führt dazu, dass Entscheidungen in diesem Modell nur zu sehr wenigen Zeitpunkten getroffen werden können. Dies wird, insbesondere für dynamische Probleme, dem zu modellierenden Problem jedoch oft nicht gerecht. Daher besteht ein weitverbreiteter Ansatz zur Bestimmung akzeptabler Lösungen darin, einen *rollierenden Planungshorizont* zu verwenden. Dazu wird das Optimierungsproblem in eine Menge von Teilproblemen mit kürzeren Zeithorizonten zerlegt und diese sukzessive gelöst. Ist das mehrstufige Problem zeitlich zerlegbar, d.h. besteht beispielsweise keine Kopplung zwischen den Entscheidungen eines Tages und jenen des Folgetages, so liefert die Optimierung mit rollierendem Horizont tatsächlich optimale Lösungen. Allerdings ist die genaue Berücksichtigung zeitkoppelnder Restriktionen, wie beispielsweise Speicherstände oder Mindestbetriebs- und -stillstandszeiten von Anlagen, problematisch.

Um den zugrunde liegenden stochastischen Prozess möglichst gut durch eine deutlich reduzierte Anzahl von Szenarien zu approximieren, wurden Methoden zur *Szenarioreduktion* entwickelt, vgl. [10]. Diese erlauben es, durch Verwendung einer geringeren Anzahl von ausgewählten Szenarien das Optimierungsproblem auch für längere Zeiträume zu lösen. Hierbei ist jedoch zu berücksichtigen, dass bei einer deutlichen Reduktion der Szenarienzahl das Modell die Unsicherheit bezüglich der stochastischen Daten zunehmend unterschätzt.

Ein weiterer Ansatz zur Reduzierung der Dimension des Optimierungsproblems besteht in der Verwendung *rekombinierender Szenariobäume*. Das hierfür bestbekannte Beispiel ist das Binomialmodell von Cox, Ross und Rubinstein [2] zur Berechnung von Optionspreisen. Für ein Problem mit T Zeitstufen verringert hier das Rekombinieren von Szenarien die Anzahl der Knoten von $2^T - 1$ auf $T(T+1)/2$ und ermöglicht so erst die Lösung von Problemen mit längeren Zeithorizonten. Rekombinierende Szenariobäume wurden des Weiteren bisher unter anderem zur Bewertung der Kosten von fluktuierender Wind-einspeisung [21] und zur Bewertung von Gasspeichern [8] eingesetzt. Allerdings ist bei der stochastischen Optimierung auf einem rekombinierenden Baum die präzise Abbildung von Nebenbedingungen, welche verschiedene Zeitstufen koppeln, nicht möglich. In [21] werden zeitkoppelnde Restriktionen daher nur näherungsweise erfüllt. In [8] wird der zulässige Bereich der gekoppelten Entscheidungsvariablen diskretisiert und ein auf

dynamischer Programmierung beruhender Lösungsansatz verwendet. Im vorliegenden Kapitel wird ein anderer Ansatz vorgestellt, der auf der folgenden Beobachtung beruht. Die *Kostenfunktion* in einem Knoten des Szenariobaums beschreibt die erwarteten, minimal erreichbaren zukünftigen Kosten in Abhängigkeit der Entscheidungen aus vorhergehenden Zeitstufen und des Wertes der stochastischen Parameter in diesem Knoten. Die erwarteten zukünftigen Kosten sind von dem im jeweiligen Knoten beginnenden Teilbaum bestimmt. Gibt es zu einem Zeitpunkt Knoten im Szenariobaum, deren Teilbäume übereinstimmen, so sind die jeweiligen Kostenfunktionen identisch. Zerlegt man das stochastische Problem zu diesen Zeitpunkten im Sinne der *Nested Benders Dekomposition* [1, 18] und approximiert die dort auftretenden Kostenfunktionen, so muss für übereinstimmende Kostenfunktionen nur eine einzige Approximation bestimmt werden. Dieses als “cut sharing” bezeichnete Prinzip wurde bereits von Infanger und Morton [13] und Pereira und Pinto [16] angewandt und führt durch die Ausnutzung von übereinstimmenden Kostenfunktionen zu einer beschleunigten Konvergenz des Lösungsalgorithmus. Allerdings gibt es mit wachsendem Zeithorizont eine exponentiell wachsende Anzahl von Knoten mit übereinstimmenden Kostenfunktionen, welche jeweils in den knotenabhängigen Entscheidungsvariablen evaluiert werden müssen. Während diese Knotenmenge in [16] auf eine zufällige Auswahl reduziert wird, wird in dem in diesem Kapitel vorgestellten Algorithmus folgende Eigenschaft genutzt: Findet man Knoten in denen die entsprechenden Entscheidungsvariablen ähnliche Werte besitzen, so genügt es die (Lipschitz-stetige) Kostenfunktion in nur einer einzigen dieser Kontrollen auszuwerten. Damit können im Verlauf des Lösungsverfahrens Knoten mit gleichem Teilbaum “rekombiniert” werden, so lange die (jeweils besten bekannten) Entscheidungen in diesen Knoten ähnlich sind. Dieses Vorgehen kann als “dynamisches Rekombinieren” von Szenarien interpretiert werden und ermöglicht, ein exponentielles Wachstum der Anzahl von Kostenfunktionsauswertungen zu verhindern. Somit lassen sich stochastische Optimierungsprobleme mit zeitkoppelnden Restriktionen, vielen Zeitstufen und einer großen Anzahl von Szenarien lösen.

Das vorliegende Kapitel gliedert sich wie folgt. In Abschnitt 10.2 werden die betrachtete Klasse von linearen mehrstufigen stochastischen Optimierungsproblemen vorgestellt und das zeitliche Dekompositionsverfahren sowie das Konzept rekombinierender Szenariobäume skizziert. In Abschnitt 10.3 wird eine Modifikation der Benders Dekomposition für Optimierungsprobleme entwickelt, die einen Szenariobaum beinhalten, in welchem Szenarien in nur einer einzelnen Zeitstufe rekombiniert werden. Diese Technik bildet die Grundlage für die Erweiterung auf den in Abschnitt 10.4 vorgestellten allgemeinen Fall mit mehrfachem Rekombinieren. Ein Algorithmus zur Konstruktion von rekombinierenden Szenariobäumen aus einer gegebenen Menge von Szenarien wird in Abschnitt 10.5 diskutiert.

10.2 Problemformulierung und zeitliche Dekomposition

10.2.1 Lineare mehrstufige stochastische Probleme

Es sei $\xi = (\xi_t)_{t=1, \dots, T}$ ein \mathbb{R}^s -wertiger zeitdiskreter stochastischen Prozess¹ mit *Zeithorizont* $T \in \mathbb{N}$ auf einem Wahrscheinlichkeitsraum $(\Omega, \mathcal{F}, \mathbb{P})$. Für $t = 1, \dots, T$ sei mit $\xi_{[t]}$ der Vektor (ξ_1, \dots, ξ_t) bezeichnet. Problem (P) stellt ein allgemeines lineares mehrstufiges stochastisches Optimierungsproblem (“MSP”) dar:

$$\min \mathbb{E} \left[\sum_{t=1}^T \langle b_t(\xi_t), x_t(\xi_{[t]}) \rangle \right] \quad (\text{P})$$

so dass $x_t : \mathbb{R}^{s \cdot t} \rightarrow X_t$ messbar, $t = 1, \dots, T$,

$$A_{t,0}(\xi_t)x_t(\xi_{[t]}) + A_{t,1}(\xi_t)x_{t-1}(\xi_{[t-1]}) = h_t(\xi_t), \quad t = 2, \dots, T.$$

Dabei bezeichnet x_t die im Zeitpunkt t zu treffenden Entscheidungen; die entsprechenden zulässigen Mengen $X_t \subset \mathbb{R}^r$ seien abgeschlossene beschränkte Polyeder. Die Entscheidung $x_t = x_t(\xi_{[t]})$ darf von den (bis zum Zeitpunkt t) aus der Beobachtung von $\xi_{[t]}$ erhaltenen Informationen abhängen. Die Abbildungen $h_t(\cdot)$, $b_t(\cdot)$ und $A_{t,0}(\cdot)$, $A_{t,1}(\cdot)$ sind vektor- bzw. matrix-wertig von geeigneter Dimension. Ohne Beschränkung der Allgemeinheit sei angenommen dass Kopplungen nur zwischen aufeinanderfolgenden Zeitschritten auftreten².

Um eine numerische Lösung von (P) zu ermöglichen wird angenommen, dass der Prozess ξ Werte in einer endlichen Menge von Szenarien annimmt und damit durch einen Szenariobaum repräsentiert werden kann, vgl. [3]. Eine Baumstruktur entsteht dabei dadurch, dass jeweils gewisse Szenarien bis zu einem Zeitpunkt $t \in \{2, \dots, T-1\}$ übereinstimmen. Die Menge aller Szenarien die jeweils bis zum Zeitpunkt t übereinstimmen, bilden einen *Knoten* des Szenariobaumes. Es sei n_t die Anzahl der Knoten zum Zeitpunkt t ; diese seien mit $i = 1, \dots, n_t$ indiziert. Der Wert des Prozesses ξ bis zum Knoten i zur Zeit t wird mit $\xi_{[t]}^{(i)}$ bezeichnet. Die *Szenarien* entsprechen den Werten des Prozesses bis zu den Endknoten, d.h. sie sind gegeben durch $\xi_{[T]}^{(i)}$, $i = 1, \dots, n_T$. Die Entscheidung $x_t(\cdot)$ zum Zeitpunkt t ist konstant auf jedem Knoten in t , insbesondere ist $x_t(\cdot)$ durch die Werte $x_t^{(i)} := x_t(\xi_{[t]}^{(i)})$, $i = 1, \dots, n_t$, definiert.

10.2.2 Dynamische Formulierung

Die diskrete Struktur des Prozesses ξ erlaubt es, Problem (P) als ein (großes) deterministisches lineares Optimierungsproblem zu formulieren. Allerdings ist eine direkte nume-

¹Im folgenden bezeichnen fett gedruckte Variablen (ξ, ξ_t, \dots) *zufällige* Größen, während die Variablen (x, x_t, \dots) für mögliche Werte dieser Größen stehen.

²Gibt es Kopplungen von nicht aufeinanderfolgenden Zeitstufen, so kann das Optimierungsproblem durch die Einführung zusätzlicher Entscheidungsvariablen auf die Form (P) gebracht werden.

rische Lösung dieses *deterministischen Äquivalents* durch verfügbare Löser für lineare Programme in der Regel nicht ohne Weiteres möglich, da mit wachsendem Zeithorizont die Anzahl der Knoten im Szenariobaum und damit die Größe des Problems (P) exponentiell wächst. Daher sind Dekompositionsverfahren notwendig, welche die besondere Struktur von (P) ausnutzen können. Verschiedene solche an stochastische Optimierungsprobleme angepasste Lösungsmethoden werden in [18] vorgestellt.

Im Folgenden wird kurz eine einfache *zeitliche* Dekomposition vorgestellt. Dazu betrachtet man eine Zeitstufe $R \in \{2, \dots, T-1\}$ und definiert die *Kostenfunktion* zur Zeit R und im Zustand $(x_R^{(i)}, \xi_{[R]}^{(i)}) \in \mathbb{R}^r \times \mathbb{R}^{s \cdot R}$ als

$$\mathcal{Q}_R(x_R^{(i)}, \xi_{[R]}^{(i)}) := \tag{Q_R}$$

$$\min \mathbb{E} \left[\sum_{t=R+1}^T \langle b_t(\xi_t), x_t(\xi_{[t]}) \rangle \middle| \xi_{[R]} = \xi_{[R]}^{(i)} \right]$$

$$\text{so dass } x_t : \mathbb{R}^{s \cdot t} \rightarrow X_t \text{ messbar,} \quad t = R+1, \dots, T,$$

$$A_{R+1,0}(\xi_{R+1})x_{R+1}(\xi_{[R+1]}) = h_{R+1}(\xi_{R+1}) - A_{R+1,1}(\xi_{R+1})x_R^{(i)}, \tag{10.1}$$

$$A_{t,0}(\xi_t)x_t(\xi_{[t]}) + A_{t,1}(\xi_t)x_{t-1}(\xi_{[t-1]}) = h_t(\xi_t), \quad t = R+2, \dots, T.$$

Dabei bezeichnet für eine Funktion f der Term $\mathbb{E}[f(\xi_{t+1}, \dots, \xi_T) | \xi_{[R]} = \xi_{[R]}^{(i)}]$ die bedingte Erwartung von $f(\xi_{t+1}, \dots, \xi_T)$, gegeben dass der Zufallsvektor $\xi_{[R]}$ den Wert $\xi_{[R]}^{(i)}$ annimmt. Die Kostenfunktion $\mathcal{Q}_R(x_R^{(i)}, \xi_{[R]}^{(i)})$ gibt die erwarteten, minimal erreichbaren Kosten an, die nach dem Zeitpunkt R anfallen, wenn bis zum Zeitpunkt R der Wert $\xi_{[R]}^{(i)}$ beobachtet und in R die Entscheidung $x_R^{(i)}$ getroffen wurde.

Mit Hilfe dieser Kostenfunktion kann das Problem (P) in der folgenden dynamischen Form geschrieben werden:

$$\min \mathbb{E} \left[\sum_{t=1}^R \langle b_t(\xi_t), x_t(\xi_{[t]}) \rangle + \mathcal{Q}_R(x_R(\xi_{[R]}), \xi_{[R]}) \right] \tag{Q_0}$$

$$\text{so dass } x_t : \mathbb{R}^{s \cdot t} \rightarrow X_t \text{ messbar,} \quad t = 1, \dots, R,$$

$$A_{t,0}(\xi_t)x_t(\xi_{[t]}) + A_{t,1}(\xi_t)x_{t-1}(\xi_{[t-1]}) = h_t(\xi_t), \quad t = 2, \dots, R.$$

Diese Formulierung hat den Vorteil, dass das Originalproblem (P) in ein Problem für die ersten R Zeitstufen (Q_0) und je ein Teilproblem (Q_R) für jede Realisierung $\xi_{[R]}^{(i)}$ von $\xi_{[R]}$ zerfällt³, vgl. Abbildung 10.1.

³Der Zufallsvektor $\xi_{[R]}$ kann die Werte $\xi_{[R]}^{(i)}$, $i = 1, \dots, n_R$, annehmen.

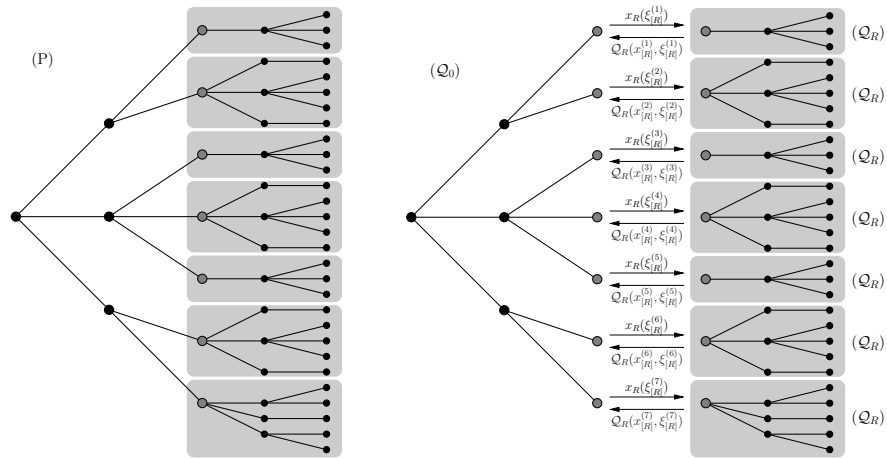


Abbildung 10.1: Zeitliche Dekomposition eines stochastischen Programmes

10.2.3 Benders Dekomposition

Innerhalb des *Benders Dekompositionsverfahrens* [1, 18, 22] werden nun die unbekannt-ten stückweise linearen konvexen Funktionen

$$Q_R(\cdot, \xi_{[R]}^{(i)}) : \mathbb{R}^r \rightarrow \mathbb{R} \cup \{+\infty\}, \quad i = 1, \dots, n_R,$$

in Problem (Q_0) sukzessive durch unterschätzende lineare Approximationen in Form von Schnittebenen ersetzt. Die so erhaltene Approximation des Problems (Q_0) ist ein lineares Optimierungsproblem und wird als *Masterproblem* bezeichnet. Die Lösung des Masterproblems liefert Entscheidungen $x_R^{(i)}$, $i = 1, \dots, n_R$ in den Endknoten des Masterproblems. Löst man nun die linearen Teilprobleme (Q_R) in diesen Punkten, d.h. berechnet man die Funktionswerte $Q_R(x_R^{(i)}, \xi_{[R]}^{(i)})$, so erhält man aus den dualen Lösungen Schnittebenen, die zur Verbesserung der Approximation von $Q_R(\cdot, \xi_{[R]}^{(i)})$ und damit zur Aktualisierung des Masterproblems genutzt werden können.

Das Masterproblem und die Probleme (Q_R) werden nun abwechselnd gelöst, bis alle nötigen Schnittebenen zur Approximation von (Q_0) bestimmt wurden. Die Entscheidungen $x_R^{(i)}$, in denen die Funktionen $Q_R(\cdot, \xi_{[R]}^{(i)})$ ausgewertet werden, stammen aus einer Lösung des Masterproblems. Durch dieses Vorgehen wird erreicht, dass die Funktionen $Q_R(\cdot, \xi_{[R]}^{(i)})$ vornehmlich dort gut approximiert werden, wo ihre Werte für die Lösung von (Q_0) tatsächlich von Interesse sind. Für eine detaillierte Darstellung dieses Vorgehensweise sei auf [18, 22] verwiesen, die Modifikation dieses Algorithmus für rekombinierende Szenariobäume wird in Abschnitt 10.3 ausführlicher erklärt.

Für lange Zeithorizonte T können die Teilprobleme (Q_R) weiter zerlegt und selbst mittels des oben beschriebenen Approximationsverfahrens gelöst werden. Dieses Vorgehen

wird als *Nested Benders Dekomposition* [1] bezeichnet. Allerdings wächst mit der Anzahl der Knoten zum Zeitpunkt R auch die Anzahl der zu approximierenden Funktionen $\mathcal{Q}_R(\cdot, \xi_{[R]}^{(i)})$, $i = 1, \dots, n_R$.

10.2.4 Rekombinierende Szenariobäume

Eine besondere Situation ergibt sich, wenn die in zwei Knoten i und j zu einem Zeitpunkt R ansetzenden Teilbäume übereinstimmen, d.h. die entsprechenden *bedingten Wahrscheinlichkeitsverteilungen* von $(\xi_{R+1}, \dots, \xi_T)$ identisch sind:

$$\mathbb{P} \left[(\xi_{R+1}, \dots, \xi_T) = a \mid \xi_{[R]} = \xi_{[R]}^{(i)} \right] = \mathbb{P} \left[(\xi_{R+1}, \dots, \xi_T) = a \mid \xi_{[R]} = \xi_{[R]}^{(k)} \right] \quad (10.2)$$

für alle $a \in \mathbb{R}^{s \cdot (T-R)}$, vgl. linke Grafik in Abbildung 10.2. Ein Ansatz zur Konstruktion von Szenariobäumen mit der Eigenschaft (10.2) wird in Abschnitt 10.5 vorgestellt.

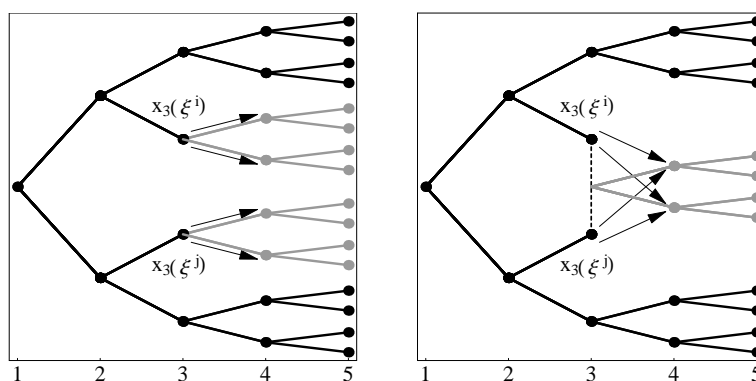


Abbildung 10.2: Szenariobaum (schematisch) mit Eigenschaft (10.2), $R = 3$ und $m_R = 2$, d.h. zwei verschiedene Teilbäume beginnen zur Zeit 3; die schwarzen und die grauen Bäume stimmen jeweils überein.

Ohne die zeitkoppelnde Restriktion (10.1) ist die Kostenfunktion $\mathcal{Q}_R(x_R^{(i)}, \xi_{[R]}^{(i)})$ unabhängig vom Wert $x_R^{(i)}$. Dann genügt es, innerhalb der Benders Dekomposition nur eines der beiden identischen Teilprobleme $\mathcal{Q}_R(x_R^{(i)}, \xi_{[R]}^{(i)})$ und $\mathcal{Q}_R(x_R^{(k)}, \xi_{[R]}^{(k)})$ zu lösen⁴. Insbesondere kann man dann den Prozess ξ durch einen Szenariobaum darstellen, in welchem die Knoten i und k zusammengelegt, d.h. *rekombiniert* werden, vgl. Abbildung 10.2. Durch wiederholtes Rekombinieren in verschiedenen Zeitstufen ergäbe sich ein Szenariobaum, dessen Knotenzahl nunmehr linear statt exponentiell mit der Anzahl der Zeitstufen wächst.

⁴Dies gilt auch, wenn die optimalen Entscheidungen $x_R^{(i)}$ und $x_R^{(k)}$ in den Knoten i und k übereinstimmen.

Da im Allgemeinen jedoch die optimalen Entscheidungen $x_R^{(i)}$ und $x_R^{(k)}$ unterschiedlich ausfallen werden, ist die Verwendung eines derart rekombinierenden Baumes zur Lösung eines Optimierungsproblems mit zeitkoppelnden Restriktionen nicht ohne Weiteres möglich. Allerdings kann Eigenschaft (10.2), d.h. die Übereinstimmung von Teilbäumen, von Nutzen sein, da diese die Identität der Kostenfunktionen $\mathcal{Q}_R(\cdot, \xi_{[R]}^{(i)})$ und $\mathcal{Q}_R(\cdot, \xi_{[R]}^{(k)})$ impliziert. Einerseits ermöglicht dies die *simultane Approximation*⁵ der Kostenfunktionen, was bereits zu einer verbesserten Konvergenz des Lösungsalgorithmus führt. Andererseits werden bei einer großen Zahl von Knoten innerhalb der Benders Dekomposition sehr viele Teilprobleme (\mathcal{Q}_R) gelöst, die sich nur in den auf der *rechten Seite* auftretenden Größen $x_R^{(i)}$ unterscheiden, vgl. (10.1). Da die Optimalwerte der Teilprobleme stetig von der rechten Seite abhängen, genügt es, eine Auswahl weniger “repräsentativer” rechter Seiten zu berücksichtigen, d.h. erfüllen i und k die Eigenschaft (10.2) und hat man ähnliche Werte $x_R^{(i)}$ und $x_R^{(k)}$, so löst man nur eines der Teilprobleme $\mathcal{Q}_R(x_R^{(i)}, \xi_{[R]}^{(i)})$ und $\mathcal{Q}_R(x_R^{(k)}, \xi_{[R]}^{(k)})$. Dieses Vorgehen wird im nächsten Abschnitt genauer beschrieben und kann als *dynamisches Rekombinieren* von Szenarien interpretiert werden.

10.3 Einfache Dekomposition mit rekombinierenden Bäumen

In diesem Abschnitt wird zunächst die Situation betrachtet, dass der Szenariobaum die Eigenschaft (10.2) in einem Zeitpunkt $R \in \{1, \dots, T-1\}$ besitzt. Es wird ein Lösungsalgorithmus vorgestellt, der diese Eigenschaft ausnutzt und das Originalproblem im Zeitpunkt R zerlegt. Im nächsten Abschnitt wird dieses Verfahren zu einer mehrfachen Dekomposition erweitert.

Die Benders Dekomposition betrachtet die dynamische Formulierung (\mathcal{Q}_0) des Problems (P) und approximiert die (konvexen und stückweise linearen) Funktionen $\mathcal{Q}_R(\cdot, \xi_{[R]}^{(i)})$, $i = 1, \dots, n_R$, von unten durch lineare Funktionen, vgl. Abbildung 10.3. Diese *Unterschätzer* werden durch das Lösen der linearen Optimierungsprobleme (\mathcal{Q}_R) bezüglich einer adaptiv gewählten Folge von Punkten $\bar{x}_R \in \mathbb{R}^r$ bestimmt. Ist dabei das Problem (\mathcal{Q}_R) für einen solchen Punkt zulässig, so lässt sich aus dessen Lösung die Ungleichung

$$\bar{q} + \langle \bar{\pi}, x_R^{(i)} - \bar{x}_R \rangle \leq \mathcal{Q}_R(x_R^{(i)}, \xi_{[R]}^{(i)}) \quad \text{für alle } x_R^{(i)} \in \mathbb{R}^r \quad (10.3)$$

ableiten. Dabei ist $\bar{q} := \mathcal{Q}_R(\bar{x}_R, \xi_{[R]}^{(i)})$ der Optimalwert des Problems (\mathcal{Q}_R) und

$$\bar{\pi} := \mathbb{E} \left[A_{R+1,1}(\boldsymbol{\xi}_{R+1})^\top \pi_{R+1}(\boldsymbol{\xi}_{R+1}) \mid \boldsymbol{\xi}_{[R]} = \xi_{[R]}^{(i)} \right],$$

wobei $\pi_{R+1}(\boldsymbol{\xi}_{R+1})$ der Teil der dualen Lösung von (\mathcal{Q}_R) ist, welcher zu der zeitkoppelnden Restriktion (10.1) gehört. Die linke Seite der Ungleichung (10.3) stellt eine lineare

⁵Das bedeutet, dass die Schnittebenen, die zur Approximation jeweils einer der Kostenfunktionen verwendet werden, ebenfalls zur Approximation der anderen dienen.

Funktion in $x_R^{(i)}$ dar und wird als *Optimalitätsschnitt* bezeichnet, da sie den Optimalwert $\mathcal{Q}_R(x_R^{(i)}, \xi_{[R]}^{(i)})$ von (\mathcal{Q}_R) (nach unten) approximiert.

Falls das Problem (\mathcal{Q}_R) für den gegebenen Referenzpunkt \bar{x}_R unzulässig ist, so lässt sich eine Ungleichung $\langle d, x_R^{(i)} \rangle \leq e$ bestimmen, welche für \bar{x}_R verletzt ist, aber für jene $x_R^{(i)}$ erfüllt ist, für die (\mathcal{Q}_R) eine zulässige Lösung besitzt, vgl. [22]. Diese Art von Ungleichung wird als *Zulässigkeitsschnitt* bezeichnet.

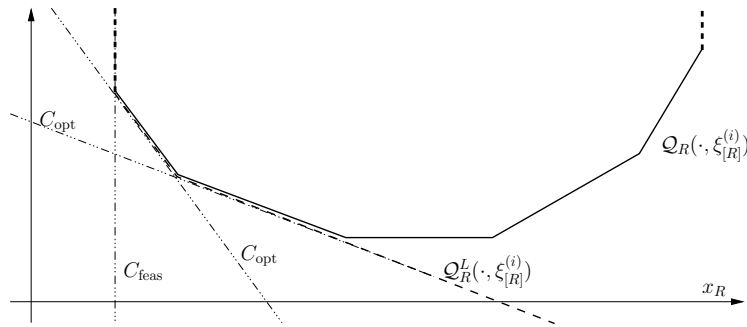


Abbildung 10.3: Approximation einer Funktion $\mathcal{Q}_R(\cdot, \xi_{[R]}^{(i)})$ durch Schnittebenen

Die Kostenfunktion $\mathcal{Q}_R(\cdot, \xi_{[R]}^{(i)})$ ist durch die bedingte Wahrscheinlichkeitsverteilung des Zufallsvektors $(\xi_{R+1}, \dots, \xi_T)$ unter der Bedingung $\{\xi_{[R]} = \xi_{[R]}^{(i)}\}$ bestimmt. Diese Verteilung ist wiederum durch den im Knoten i zur Zeit R ansetzenden Teilbaum des Szenariobaums festgelegt. Ist der den Prozess ξ beschreibende Szenariobaum nun derart, dass einige der in Zeitstufe R ansetzenden Teilbäume übereinstimmen, also die Bedingung (10.2) für zwei Knoten $i, k \in \{1, \dots, n_R\}$ erfüllt ist, so stimmen auch die Funktionen $\mathcal{Q}_R(\cdot, \xi_{[R]}^{(i)})$ und $\mathcal{Q}_R(\cdot, \xi_{[R]}^{(k)})$ überein und können simultan approximiert werden. Dies bedeutet, dass jede die Funktion $\mathcal{Q}_R(\cdot, \xi_{[R]}^{(i)})$ approximierende Schnittebene ebenfalls zur Approximation von $\mathcal{Q}_R(\cdot, \xi_{[R]}^{(k)})$ verwendet werden kann und umgekehrt, vgl. Abbildung 10.4. Um dies zu formalisieren sei eine (nicht-surjektive) Abbildung $\lambda : \{1, \dots, n_R\} \rightarrow \{1, \dots, n_R\}$ definiert, die jedem Knoten i zur Zeit R einen (beliebigen) repräsentativen Knoten k mit dem gleichen Teilbaum zuordnet:

$$\lambda(i) := k \quad \text{für alle } i \in \{1, \dots, n_R\}, \text{ die mit } k \text{ Bedingung (10.2) erfüllen.} \quad (10.4)$$

Der Wert des Prozesses ξ in dem repräsentativen Knoten $\lambda(i)$ zur Zeit R wird im Folgenden mit $\lambda(\xi_{[R]}^{(i)}) := \xi_{[R]}^{\lambda(i)}$ bezeichnet.

Für gegebene Mengen von Optimalitätsschnitten $C_{\text{opt}}(\lambda(\xi_{[R]}^{(i)}))$ und Zulässigkeitsschnitten $C_{\text{feas}}(\lambda(\xi_{[R]}^{(i)}))$ für die Funktion $\mathcal{Q}_R(\cdot, \xi_{[R]}^{(i)})$ definiert man das folgende *Masterpro-*

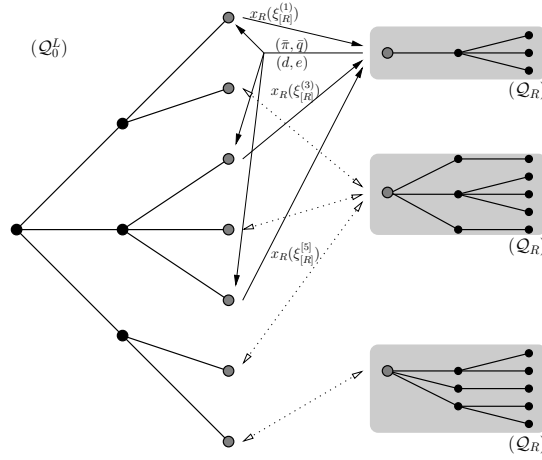


Abbildung 10.4: Zeitliche Dekomposition mit rekombinierendem Szenariobaum

blem⁶, vgl. [1, 18, 22]:

$$\min_{x, w_R} \mathbb{E} \left[\sum_{t=1}^R \langle b_t(\xi_t), x_t(\xi_{[t]}) \rangle + w_R(\xi_{[R]}) \right] \quad (\mathcal{Q}_0^L)$$

so dass $x_t : \mathbb{R}^{s \cdot t} \rightarrow X_t$ messbar, $w_R : \mathbb{R}^{s \cdot R} \rightarrow \mathbb{R}$ messbar, $t = 1, \dots, R$,

$$A_{t,0}(\xi_t)x_t(\xi_{[t]}) + A_{t,1}(\xi_t)x_{t-1}(\xi_{[t-1]}) = h_t(\xi_t), \quad t = 2, \dots, R,$$

$$\langle d, x_R(\xi_{[R]}) \rangle \leq e, \quad (d, e) \in C_{\text{feas}}(\lambda(\xi_{[R]})),$$

$$\bar{q} + \langle \bar{\pi}, x_R(\xi_{[R]}) - \bar{x}_R \rangle - w_R(\xi_{[R]}) \leq 0, \quad (\bar{x}_R, \bar{q}, \bar{\pi}) \in C_{\text{opt}}(\lambda(\xi_{[R]})).$$

Da die (unbekannten) Funktionen $\mathcal{Q}_R(\cdot, \xi_{[R]}^{(i)})$ hier durch lineare Unterschätzerfunktionen ersetzt wurden, liefert die Lösung von Problem (\mathcal{Q}_0^L) eine untere Schranke an den Optimalwert von Problem (P).

Wie in Abschnitt 10.2.3 beschrieben, passt der in Abbildung 10.5 dargestellte Lösungsalgorithmus durch Einfügen von Schnittebenen in $C_{\text{opt}}(\cdot)$ und $C_{\text{feas}}(\cdot)$ das Masterproblem (\mathcal{Q}_0^L) adaptiv an das Originalproblem (\mathcal{Q}_0) an. Insbesondere werden dabei beim Lösen von Teilproblemen $\mathcal{Q}_R(x_R^{(k)}, \xi_{[R]}^{(i)})$ und $\mathcal{Q}_R(x_R^{(k)}, \xi_{[R]}^{(k)})$ Schnittebenen in dieselben Mengen $C_{\text{opt}}(\lambda(\xi_{[R]}^{(i)}))$ und $C_{\text{feas}}(\lambda(\xi_{[R]}^{(i)}))$ eingefügt. Durch dieses simultane Aufdatieren kann bereits eine deutliche Beschleunigung des klassischen Benders Algorithmus erreicht werden.

⁶Die Schreibweise $\langle d, x_R(\xi_{[R]}) \rangle \leq e$ für $(d, e) \in C_{\text{feas}}(\lambda(\xi_{[R]}))$ in den Nebenbedingungen von (\mathcal{Q}_0^L) bedeutet, dass diese Bedingung für alle möglichen Realisierungen von $\xi_{[R]}$ gestellt wird, d.h. es wird $\langle d, x_R(\xi_{[R]}^{(i)}) \rangle \leq e$ für $(d, e) \in C_{\text{feas}}(\lambda(\xi_{[R]}^{(i)}))$ und $i = 1, \dots, n_R$ gefordert. Analog gilt dies für die Optimalitätsschnitte.

Falls die Anzahl der Knoten n_R sehr groß ist, so muss auch eine sehr große Zahl verschiedener Teilprobleme $Q_R(x_R^{(i)}, \xi_{[R]}^{(i)})$, $i = 1, \dots, n_R$, gelöst werden. Eigenschaft (10.2) führt nun aber dazu, dass nur noch wenige *verschiedene* Teilprobleme $Q_R(\cdot, \lambda(\xi_{[R]}^{(i)}))$ jeweils in einer sehr großen Anzahl von Punkten $x_R^{(i)}$ betrachtet werden müssen. Da nun der Wert $Q_R(x_R^{(i)}, \lambda(\xi_{[R]}^{(i)}))$ stetig von $x_R^{(i)}$ abhängt, ist es naheliegend, sich auf die Berechnung von $Q_R(\cdot, \lambda(\xi_{[R]}^{(i)}))$ in einer reduzierten Auswahl von Punkten zu beschränken. Diese Selektion wird in Schritt 4 des in Abbildung 10.5 dargestellten Algorithmus durchgeführt; die Genauigkeit dieser Operation wird durch den Parameter ρ gesteuert und im Verlauf der Optimierung sukzessive erhöht. Somit werden die Teilprobleme in Knoten i und k während einer Iteration des Algorithmus' identifiziert, falls sich die zeitkoppelnden Variablen $x_R^{(i)}$ und $x_R^{(k)}$ in diesem Iterationsschritt um weniger als den Wert ρ unterscheiden. Dieses Vorgehen kann als *dynamisches Rekombinieren von Szenarien* interpretiert werden.

1. Setze $C_{\text{feas}}(\cdot) := \emptyset$, $C_{\text{opt}}(\cdot) := \{(0, L, 0)\}$ und $\rho := \rho_{\text{start}}$, wobei die Konstante L eine untere Schranke an $Q_R(\cdot, \cdot)$ ist.
2. Löse das Masterproblem (Q_0^L) .
3. Falls (Q_0^L) unzulässig ist, so stoppe: (Q_0) ist unzulässig.
Andernfalls sei (\bar{x}, \bar{w}) eine Lösung von (Q_0^L) .
4. Für alle $\xi_{[R]}^{(i)}$, $i = 1, \dots, n_R$: Falls *kein* $x_R^{(i)}$ mit $\|x_R^{(i)} - \bar{x}_R(\xi_{[R]}^{(i)})\| \leq \rho$ existiert, für welches $Q_R(x_R^{(i)}, \lambda(\xi_{[R]}^{(i)}))$ bereits bestimmt wurde:
 - (a) Bestimme $Q_R(\bar{x}_R(\xi_{[R]}^{(i)}), \lambda(\xi_{[R]}^{(i)}))$ durch Lösung von (Q_R) .
 - (b) Falls (Q_R) unzulässig ist, so konstruiere einen Zulässigkeitsschnitt und füge diesen zu $C_{\text{feas}}(\lambda(\xi_{[R]}^{(i)}))$ hinzu.
 - (c) Falls (Q_R) zulässig ist und $Q_R(\bar{x}_R(\xi_{[R]}^{(i)}), \lambda(\xi_{[R]}^{(i)})) > \bar{w}_R(\xi_{[R]}^{(i)})$, so konstruiere einen Optimalitätsschnitt und füge diesen zu $C_{\text{opt}}(\lambda(\xi_{[R]}^{(i)}))$ hinzu.
5. Falls Optimalitäts- oder Zulässigkeitsschnitte generiert wurden, gehe zurück zu 2.
Sonst: Falls $\rho > \rho_{\text{end}}$, verringere ρ und gehe zurück zu 2.
Sonst: Falls $\rho \leq \rho_{\text{end}}$, stoppe. Problem (Q_0) wurde (ρ_{end} -) optimal gelöst.

Abbildung 10.5: Lösung von (Q_0) mittels einer modifizierten Benders Dekomposition.

Der Lösungsalgorithmus stoppt wenn das Masterproblem (Q_0^L) unzulässig ist oder die Generierung von Schnittebenen mit minimalem $\rho = \rho_{\text{end}}$ beendet ist. Im ersten Fall

bedeutet dies die Unzulässigkeit des Originalproblems (\mathcal{Q}_0), während im zweiten Fall eine (ρ_{end} -) optimale Lösung von (\mathcal{Q}_0) bestimmt wurde, vgl. Proposition 10.4.1. Andere Stoppkriterien, welche es erlauben den Algorithmus mit $\rho > \rho_{\text{end}}$ zu beenden sobald der Approximationsfehler unter eine vorgegebene Toleranz fällt, werden in Abschnitt 10.4.2 diskutiert.

10.4 Mehrfache Dekomposition mit rekombinierenden Bäumen

Besitzt der den Prozess ξ beschreibende Szenariobaum die Eigenschaft (10.2) in mehreren Zeitpunkten R_1, \dots, R_n mit

$$0 =: R_0 < R_1 < \dots < R_n < R_{n+1} := T,$$

so lässt sich der im letzten Abschnitt vorgestellte Dekompositionsalgorithmus rekursiv erweitern.

Wie in Abschnitt 10.2.2 definiert man dazu Kostenfunktionen durch $\mathcal{Q}_{R_{n+1}} \equiv 0$ und

$$\mathcal{Q}_{R_j}(x_{R_j}^{(i)}, \xi_{[R_j]}^{(i)}) := \min \mathbb{E} \left[\sum_{t=R_j+1}^{R_{j+1}} \langle b_t(\xi_t), x_t(\xi_{[t]}) \rangle + \mathcal{Q}_{R_{j+1}}(x_{R_{j+1}}(\xi_{[R_{j+1}]}), \xi_{[R_{j+1}]}) \mid \xi_{[R_j]} = \xi_{[R_j]}^{(i)} \right] \quad (\mathcal{Q}_{R_j})$$

so dass $x_t : \mathbb{R}^{s \cdot t} \rightarrow X_t$ messbar,

$$t = R_j + 1, \dots, R_{j+1},$$

$$A_{R_j+1,0}(\xi_{R_j+1})x_{R_j+1}(\xi_{[R_j+1]}) = h_{R_j+1}(\xi_{R_j+1}) - A_{R_j+1,1}(\xi_{R_j+1})x_{R_j}^{(i)}$$

$$A_{t,0}(\xi_t)x_t(\xi_{[t]}) + A_{t,1}(\xi_t)x_{t-1}(\xi_{[t-1]}) = h_t(\xi_t), \quad t = R_j + 2, \dots, R_{j+1},$$

für $j = 0, \dots, n$. Wieder stimmen die Funktionen $\mathcal{Q}_{R_j}(\cdot, \xi_{[R_j]}^{(i)})$ und $\mathcal{Q}_{R_j}(\cdot, \xi_{[R_j]}^{(k)})$ überein, falls die Knoten i und k zur Zeitstufe R_j die Bedingung (10.2) mit $R = R_j$ erfüllen. Wie in Definition (10.4) seien für $j = 1, \dots, n$ Abbildungen λ_j definiert, die jedem Knoten i im Zeitpunkt R_j einen repräsentativen Knoten k mit dem gleichen Teilbaum zuordnen. Es sei m_j die Anzahl *verschiedener* Teilbäume, die im Zeitpunkt R_j ansetzen. Für $\bar{i} = 1, \dots, m_j$ bezeichnet $\lambda_j^{(\bar{i})} := \xi_{[R_j]}^{(\lambda_j(i))}$ den Wert von ξ bis zum \bar{i} -ten repräsentativen Knoten⁷ zum Zeitpunkt R_j .

⁷Der Wert i in der Definition von $\lambda_j^{(\bar{i})}$ bezeichnet einen beliebigen Knoten zum Zeitpunkt R_j , der ebenfalls den \bar{i} -ten Teilbaum besitzt, d.h. $\lambda_j(i) = \bar{i}$.

10.4.1 Lösung mit mehrfach rekombinierenden Bäumen

Analog zu (\mathcal{Q}_0^L) seien für $j = 0, \dots, n$ und $i = 1, \dots, m_j$ die Masterprobleme

$$\mathcal{Q}_{R_j}^L(x_{R_j}^{(i)}, \xi_{[R_j]}^{(i)}) := \min_{x, w_{R_{j+1}}} \mathbb{E} \left[\sum_{t=R_j+1}^{R_{j+1}} \langle b_t(\xi_t), x_t(\xi_{[t]}) \rangle + w_{R_{j+1}}(\xi_{[R_{j+1}]}) \mid \xi_{[R_j]} = \xi_{[R_j]}^{(i)} \right] \quad (\mathcal{Q}_{R_j}^L)$$

so dass $x_t : \mathbb{R}^{s \cdot t} \rightarrow X_t$, $w_{R_{j+1}} : \mathbb{R}^{s \cdot R_{j+1}} \rightarrow \mathbb{R}$ messbar, $t = R_j + 1, \dots, R_{j+1}$,

$$A_{R_{j+1},0}(\xi_{R_{j+1}})x_{R_{j+1}}(\xi_{[R_{j+1}]}) = h_{R_{j+1}}(\xi_{R_{j+1}}) - A_{R_{j+1},1}(\xi_{R_{j+1}})x_{R_j}^{(i)},$$

$$A_{t,0}(\xi_t)x_t(\xi_{[t]}) + A_{t,1}(\xi_t)x_{t-1}(\xi_{[t-1]}) = h_t(\xi_t), \quad t = R_j + 1, \dots, R_{j+1},$$

$$\langle d, x_{R_{j+1}}(\xi_{[R_{j+1}]}) \rangle \leq e, \quad (d, e) \in C_{\text{feas}}(\lambda_{j+1}(\xi_{[R_{j+1}]})),$$

$$\bar{q} + \langle \bar{\pi}, x_{R_{j+1}}(\xi_{[R_{j+1}]}) - \bar{x}_{R_{j+1}} \rangle - w_{R_{j+1}}(\xi_{[R_{j+1}]}) \leq 0,$$

$$(\bar{x}_{R_{j+1}}, \bar{q}, \bar{\pi}) \in C_{\text{opt}}(\lambda_{j+1}(\xi_{[R_{j+1}]})),$$

definiert, wobei $C_{\text{opt}}(\lambda_{j+1}^{(\bar{i})})$ und $C_{\text{feas}}(\lambda_{j+1}^{(\bar{i})})$, $\bar{i} = 1, \dots, m_{j+1}$, Mengen von Optimalitäts- bzw. Zulässigkeitschnitten sind, welche die Funktion $\mathcal{Q}_{R_{j+1}}^L(\cdot, \lambda_{j+1}^{(\bar{i})})$ approximieren⁸, siehe auch [15].

Der Lösungsalgorithmus löst die Masterprobleme $(\mathcal{Q}_{R_j}^L)$, $j = 0, \dots, n$, abwechselnd in einem Vorwärts- und einem Rückwärtsmodus, vgl. auch Abbildung 10.6. Der *Vorwärtsmodus* dient der Weitergabe primaler Information in Form von Kontrollpunkten an zukünftige Zeitstufen. Dabei werden nacheinander die Masterprobleme in den Zeitstufen R_0, \dots, R_n gelöst. Dabei liefert das Lösen der Probleme zum Zeitpunkt R_{j-1} Mengen $Z_j^{(\bar{i})}$ von Paaren $(x_{R_j}^{(\bar{i})}, w_{R_j}^{(\bar{i})})$, wobei $w_{R_j}^{(\bar{i})}$ der Wert der Schnittebenenapproximation von $\mathcal{Q}_{R_j}^L(\cdot, \lambda_j^{(\bar{i})})$ in der Entscheidungsvariablen $x_{R_j}^{(\bar{i})}$ ist. Im Zeitpunkt R_j werden die Funktionen $\mathcal{Q}_{R_j}^L(\cdot, \lambda_j^{(\bar{i})})$, $\bar{i} = 1, \dots, m_j$, für die Werte $x_{R_j}^{(\bar{i})}$ aus $Z_j^{(\bar{i})}$ berechnet, d.h. die Probleme $(\mathcal{Q}_{R_j}^L)$ gelöst. Falls nun $w_{R_j}^{(\bar{i})} < \mathcal{Q}_{R_j}^L(x_{R_j}^{(\bar{i})}, \lambda_j^{(\bar{i})})$, so ist die Approximation an der Stelle $x_{R_j}^{(\bar{i})}$ noch nicht exakt und ein Optimalitäts- oder Zulässigkeitschnitt wird aus der Lösung von $(\mathcal{Q}_{R_j}^L)$ generiert, vgl. Abschnitt 10.3 bzw. [22]. Falls $(\mathcal{Q}_{R_j}^L)$ zulässig war, so werden die Lösungen $x_{R_{j+1}}(\cdot)$ zusammen mit den Werten $w_{R_{j+1}}(\cdot)$ aus der Lösung von $(\mathcal{Q}_{R_j}^L)$ den Mengen $Z_{j+1}^{(\bar{k})}$, $\bar{k} = 1, \dots, m_{j+1}$, für den nächsten Zeitschritt R_{j+1} zugefügt. Falls ein Teilproblem unzulässig war, d.h. der zugehörige Kontrollpunkt durch einen Zulässigkeitschnitt in der vorhergehenden Zeitperiode abgeschnitten werden muss, so wechselt der Algorithmus kurzfristig in den Rückwärtsmodus.

Der *Rückwärtsmodus* dient der Rückgabe dualer Information in Form von Optimalitäts- und Zulässigkeitschnitten an zurückliegende Zeitstufen. Dabei werden nacheinander die Masterprobleme $(\mathcal{Q}_{R_j}^L)$ der Zeitstufen R_n, \dots, R_0 in den im Vorwärtsmodus erzeugten

⁸Für $j = n$ ist $\mathcal{Q}_{R_{j+1}}^L(\cdot, \cdot) \equiv 0$, d.h. es sei $C_{\text{feas}}(\cdot) := \emptyset$ und $C_{\text{opt}}(\cdot) := \{(0, 0, 0)\}$ in $(\mathcal{Q}_{R_n}^L)$.

Punkten $x_{R_j}^{(i)} \in Z_j^{(i)}$ gelöst. Die Lösung eines Masterproblems zur Zeit R_j aktualisiert dabei gegebenenfalls die im Vorwärtsmodus bestimmten Schnittebenen. Die aktualisierten Schnitte werden zu $C_{\text{opt}}(\lambda_j^{(i)})$ bzw. $C_{\text{feas}}(\lambda_j^{(i)})$ hinzugefügt, d.h. die Masterprobleme zur Zeit R_{j-1} werden aufdatiert.

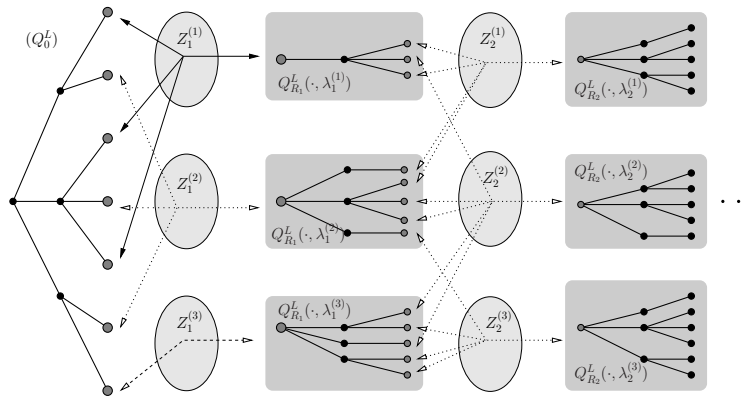


Abbildung 10.6: Schematische Darstellung des Dekompositionsalgorithmus für mehrfach rekombinierende Szenariobäume

Der Algorithmus ist beendet wenn entweder das Masterproblem (Q_0^L) in der ersten Zeitstufe unzulässig ist, oder alle Masterprobleme zur Optimalität gelöst werden konnten und keine Schnittebenen mehr generiert werden. Im ersten Fall ist auch das Originalproblem (P) unzulässig, im zweiten Fall wurde es zur Optimalität gelöst. Eine detaillierte Beschreibung des Nested Benders Algorithmus findet man in den Arbeiten [1, 9, 18].

Im allgemeinen Fall entspricht die Anzahl der Teilprobleme zur Zeit R_j der Anzahl der Knoten n_{R_j} des Baumes zum Zeitpunkt R_j . Im betrachteten Fall übereinstimmender Teilbäume gibt es nur m_j verschiedene Teilprobleme, und damit auch nur m_j zu approximierende Kostenfunktionen. Daher kann so eine deutlich schnellere Konvergenz des Lösungsalgorithmus erreicht werden. Allerdings muss jedes der m_j verschiedenen Teilprobleme wiederum für alle Punkte $x_{R_j}^{(i)} \in Z_j^{(i)}$ gelöst werden. Wie in Abschnitt 10.3 erlaubt es die Stetigkeit der Funktionen $Q_{R_j}^L(\cdot, \lambda_j^{(i)})$ wieder, die Mengen $Z_j^{(i)}$ derart zu reduzieren, dass die verbleibenden Punkte paarweise jeweils einen Abstand größer als ein Parameter ρ zueinander haben.

Zu Beginn des Lösungsprozesses sind noch keine Informationen (in Form von Schnittebenen) über die Kostenfunktionen in den Masterproblemen bekannt. Daher können hier Punkte $x_{R_j}^{(i)}$ generiert werden, die noch weit von einer optimalen Lösung entfernt liegen. Wählt man zu Beginn nun einen großen Wert für ρ , so werden nur wenige dieser Punkte ausgewertet und die Kostenfunktionen fernab der optimalen Lösung nicht unnötig genau approximiert. Hat man den Baum einige Male mit hohem Wert von ρ durchlaufen, so

liefern die Masterprobleme bereits eine grobe Darstellung der Kostenfunktionen und die im Folgenden generierten Kontrollpunkte befinden sich bereits in der Nähe der optimalen Lösung. Reduziert man nun den Parameter ρ sukzessive, so werden die Approximationen der Kostenfunktionen in der Nähe der optimalen Lösung weiter verfeinert. Die Wahl des letzten, minimalen ρ -Wertes, für welchen Kostenfunktionen ausgewertet und Schnittebenen generiert werden, bestimmt damit auch die Güte der approximierten Lösung:

Proposition 10.4.1 ([15]). Es sei v der Optimalwert von (P) und v^L der Optimalwert der Approximation (Q_0^L) . Ist (Q_{R_j}) lösbar⁹ für alle $x_{R_j}^{(\bar{i})} \in X_{R_j}$, so gilt die folgende Aussage: Der Nested Benders Dekompositions-Algorithmus mit Reduktion der Mengen $Z_j^{(\bar{i})}$, $\bar{i} = 1, \dots, m_j$, stoppt nach endlich vielen Schritten mit einer Lösung von (Q_0^L) die ρ -optimal für (P) ist, d.h. es existiert eine Konstante $C \geq 0$, so dass

$$|v - v^L| \leq C\rho. \quad (10.5)$$

10.4.2 Ein adaptives Abbruchkriterium durch Verwendung oberer Schranken

Die Wahl des Parameters ρ erlaubt eine Steuerung der Approximationsgüte und hat einen erheblichen Einfluss auf die Laufzeit des Lösungsalgorithmus [15]. Einerseits zeigt Proposition 10.4.1, dass für hinreichend kleine Werte von ρ die Genauigkeit der Approximation beliebig gut wird. Andererseits ist der Wert der Größe C in (10.5) im Allgemeinen nicht bekannt und der tatsächliche Approximationsfehler mitunter deutlich kleiner als $C\rho$. Daher lässt sich aus der Abschätzung (10.5) im Allgemeinen nicht ableiten, wie ρ gewählt werden sollte um eine festgelegte Approximationsgüte zu garantieren.

Im Folgenden wird eine Möglichkeit studiert den Approximationsfehler $Q_{R_j}(\cdot, \lambda_j^{(\bar{i})}) - Q_{R_j}^L(\cdot, \lambda_j^{(\bar{i})})$ und damit den Term $|v - v^L|$ während des Lösens abzuschätzen. Diese besteht darin, zusätzlich zu den *unteren* Approximationen $Q_{R_j}^L(\cdot, \lambda_j^{(\bar{i})})$ auch *obere* Approximationen $Q_{R_j}^U(\cdot, \lambda_j^{(\bar{i})})$ zu bestimmen. Solche oberen Schranken lassen sich rekursiv wie folgt bestimmen. Es sei $Y_{j+1}(\lambda_{j+1}^{(\bar{i})})$ eine Menge von Punkten $y \in X_{R_{j+1}}$ und Werten \bar{q}_y^U , von denen bekannt ist, dass sie den Wert $Q_{R_{j+1}}(y, \lambda_{j+1}^{(\bar{i})})$ überschätzen. Aufgrund der Konvexität von $Q_{R_{j+1}}(\cdot, \lambda_{j+1}^{(\bar{i})})$ lässt sich dann mittels Interpolationen der Werte \bar{q}_y^U eine Funktion konstruieren, die $Q_{R_{j+1}}(\cdot, \lambda_{j+1}^{(\bar{i})})$ nach oben abschätzt, vgl. Abbildung 10.7. Ersetzt man im Masterproblem (Q_{R_j}) die Kostenfunktion $Q_{R_{j+1}}(\cdot, \lambda_{j+1}^{(\bar{i})})$ durch eine solche interpolierende Funktion, so erhält man wiederum eine obere Schranke an $Q_{R_j}(\cdot, \lambda_j^{(\bar{i})})$. Formal sei dazu zunächst $Q_{R_n}^U(\cdot, \lambda_n^{(\bar{i})}) := Q_{R_n}(\cdot, \lambda_n^{(\bar{i})})$, $\bar{i} = 1, \dots, m_n$, definiert. Für

⁹Diese Voraussetzung dient hier zur Vereinfachung und ist bei geeigneter Reduzierung der Mengen $Z_j^{(\bar{i})}$ nicht notwendig, vgl. [15].

$j = 0, \dots, n - 1$ sei weiterhin

$$\mathcal{Q}_{R_j}^U(x_{R_j}^{(\bar{i})}, \lambda_j^{(\bar{i})}) := \min_{x, \alpha} \mathbb{E} \left[\sum_{t=R_j+1}^{R_{j+1}} \langle b_t(\xi_t), x_t(\xi_{[t]}) \rangle + \sum_{y \in Y_{j+1}(\lambda_{j+1}(\xi_{[R_{j+1}]})} \alpha_y(\xi_{[R_{j+1}]} \bar{q}_y^U \middle| \xi_{[R_j]} = \lambda_j^{(\bar{i})} \right] \quad (\mathcal{Q}_{R_j}^U)$$

so dass $x_t : \mathbb{R}^{s \cdot t} \rightarrow X_t$ messbar,

$$t = R_j + 1, \dots, R_{j+1},$$

$$A_{R_{j+1},0}(\xi_{R_{j+1}})x_{R_{j+1}}(\xi_{[R_{j+1}]}) = h_{R_{j+1}}(\xi_{R_{j+1}}) - A_{R_{j+1},1}(\xi_{R_{j+1}})x_{R_j}^{(\bar{i})},$$

$$A_{t,0}(\xi_t)x_t(\xi_{[t]}) + A_{t,1}(\xi_t)x_{t-1}(\xi_{[t-1]}) = h_t(\xi_t), \quad t = R_j + 2, \dots, R_{j+1},$$

$\alpha_y : \mathbb{R}^{s \cdot R_{j+1}} \rightarrow \mathbb{R}_+$ messbar,

$$y \in Y_{j+1}(\lambda_{j+1}(\xi_{[R_{j+1}]})$$

$$x_{R_{j+1}}(\xi_{[R_{j+1}]}) = \sum_{y \in Y_{j+1}(\lambda_{j+1}(\xi_{[R_{j+1}]})} \alpha_y(\xi_{[R_{j+1}]}) y,$$

$$\sum_{y \in Y_{j+1}(\lambda_{j+1}(\xi_{[R_{j+1}]})} \alpha_y(\xi_{[R_{j+1}]}) = 1,$$

wobei die Mengen $Y_{j+1}(\lambda_{j+1}^{(\bar{i})})$, $\bar{i} = 1, \dots, m_{j+1}$, aus Kontrollpunkten (oder Stützstellen) $y \in Y_{j+1}(\lambda_j^{(\bar{i})})$ bestehen, für die der Wert $\bar{q}_y^U := \mathcal{Q}_{R_{j+1}}^U(y, \lambda_{j+1}^{(\bar{i})})$, bereits berechnet wurde. Die Wahl der $\alpha_y(\xi_{[R_{j+1}]})$ zwingen die Kontrollpunkte $x_{R_{j+1}}(\xi_{[R_{j+1}]})$ einen Punkt in der konvexen Hülle von $Y_{j+1}(\lambda_{j+1}(\xi_{[R_{j+1}]})$ anzunehmen. Die Kosten dieses Punktes im Sinne der Kostenfunktion $\mathcal{Q}_{R_{j+1}}^U(\cdot, \lambda_{j+1}(\xi_{[R_{j+1}]})$ werden dann aus der zugehörigen Konvexkombination der bekannten Werte \bar{q}_y^U nach oben abgeschätzt.

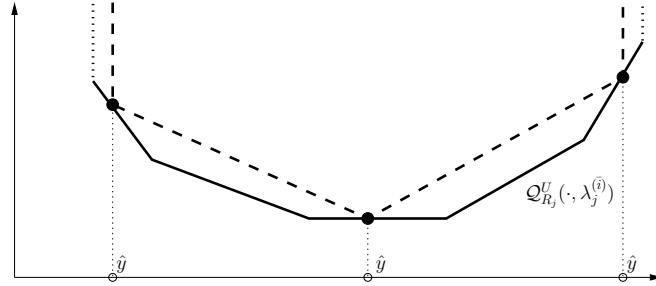


Abbildung 10.7: Obere Schranken (gestrichelt) an Kostenfunktionen mittels Konvexkombination aus bereit bestimmten Werten in Punkten \hat{y} .

Die Stützstellen $Y_j(\lambda_j^{(\bar{i})})$, $\bar{i} = 1, \dots, m_j$, werden analog zu den Optimalitäts- und Zulässigkeitschnitten gebildet. Zu Beginn des Lösungsprozesses sind die Mengen $Y_j(\lambda_j^{(\bar{i})})$ leer, wodurch $(\mathcal{Q}_{R_j}^U)$ für $j < n$ den Wert $+\infty$ annimmt. Während der Algorithmus den Baum im Vorwärts- und Rückwärtsmodus durchläuft, werden die Probleme $(\mathcal{Q}_{R_j}^U)$ für verschiedene $x_{R_j}^{(\bar{i})} \in Z_j^{(\bar{i})}$ gelöst. Liefert $(\mathcal{Q}_{R_j}^U)$ eine zulässig Lösung, so wird $x_{R_j}^{(\bar{i})}$ zu $Y_j(\lambda_j^{(\bar{i})})$ hinzugefügt und damit die Approximationen zur Zeitstufe R_{j-1} verbessert.

Die Approximationen $(Q_{R_j}^U)$ liefern sowohl ein ‐lokales‐ (d.h. auf einzelne Knoten bezogenes) als auch ein ‐globales‐ (d.h. auf den gesamten Baum bezogenes) Abbruchkriterium: Wurde für einen Kontrollpunkt $x_{R_j}^{(i)}$ sowohl die untere Schranke $Q_{R_j}^L(x_{R_j}^{(i)}, \lambda_j^{(i)})$ als auch die obere Schranke $Q_{R_j}^U(x_{R_j}^{(i)}, \lambda_j^{(i)})$ bestimmt und liegen diese nahe genug beieinander, so müssen die Kontrollpunkte $x_{R_{j+1}}(\cdot)$ aus der Lösung von $(Q_{R_j}^L)$ nicht mehr an die nächste Zeitperiode weitergegeben werden, da der zu approximierende (unbekannte) Wert $Q_{R_j}(x_{R_j}^{(i)}, \lambda_j^{(i)})$ zwischen der unteren und oberen Schranke liegen muß. Bezeichnet v^U den Optimalwert von $(Q_{R_0}^U)$, so erhält man die Fehlerabschätzung

$$v - v^L \leq v^U - v^L.$$

Dies bedeutet, dass der Lösungsalgorithmus beendet werden kann, sobald der Wert $v^U - v^L$ eine gegebene Toleranz unterschreitet. Abhängig vom Problem kann dies bereits für relativ große Werte von ρ erreicht werden, vgl. [15].

10.5 Konstruktion rekombinierender Szenariobäume

In diesem Abschnitt wird eine Methode zur Konstruktion von Szenariobäumen vorgestellt, welche die Eigenschaft (10.2) für gewisse Knoten zu verschiedenen Zeitstufen R_1, \dots, R_n besitzen. Ausgehend von einer Menge von Szenarien, die beispielsweise aus historischen Daten oder einer Simulation stammen können, wird durch stufenweises Zusammenfassen der Szenarien zu Knoten der Baum konstruiert. In diesem Sinne ist die vorgestellte Methode eng verwandt mit dem in [12] entwickelten Verfahren.

Die im Zeitpunkt $t = 1$ übereinstimmenden N Ausgangs-Szenarien seien mit $\zeta^{i'} = (\zeta_1^{i'}, \dots, \zeta_T^{i'})$, $i' = 1, \dots, N$, bezeichnet. In dem in Abbildung 10.8 beschriebenen Algorithmus werden einerseits für jede Zeitperiode $[R_j + 1, R_{j+1}]$ nicht-rekombinierende Szenariobäume konstruiert (Schritt 2), andererseits werden zum Zeitpunkt R_{j+1} Knoten verschiedener Teilbäume mit ähnlichen Werten identifiziert. Diese Knoten erhalten dann den gleichen Teilbaum für die folgende Zeitperiode $[R_{j+1} + 1, R_{j+2}]$ (Schritt 1). Die verwendete Notation wird in Tabelle 10.1 erklärt. Die Werte m_j , $n_j(h)$, und $s_{t+1}(i)$, welche die Struktur des Szenariobaumes bestimmen, können entweder a priori festgelegt, oder innerhalb des Algorithmus so bestimmt werden, dass gewisse Fehlerschranken eingehalten werden.

Die Bestimmung der Mengen A in den Schritten 1a und 2a,b sind sogenannte *k-mean clustering Probleme*, welche man mit Hilfe der in [4, 11] vorgeschlagenen *forward selection* Heuristik approximativ lösen kann.

Initialisierung: Setze $C_1^{(1,1)} := \underline{C}_{R_0}^{(1)} := \{1, \dots, N\}$, $\xi_1^{(1)} := \zeta_1^1$ und $\underline{C}_{R_j}^{(h)} := C_t^{(h,i)} := \emptyset$ für alle h, j, i .

Für $j = 0, \dots, n$ (**Konstruktion für den Zeitraum** $[R_j + 1, R_{j+1}]$)

1. Falls $j > 0$: (**Rekombinieren.**)

(a) Finde eine Menge $A = \{a_1, \dots, a_{m_j}\} \subset \{1, \dots, N\}$, so dass der folgende Abstand aller Knoten (h, i) zur Zeit R_j zum nächsten Element ζ^{a_l} klein ist:

$$\sum_{\substack{h=1, \dots, m_{j-1} \\ i=1, \dots, n_{R_j}(h)}} \min_{a_l \in A} \sum_{i' \in C_{R_j}^{(h,i)}} \|(\zeta_{R_j-\tau}^{i'}, \dots, \zeta_{R_j}^{i'}) - (\zeta_{R_j-\tau}^{a_l}, \dots, \zeta_{R_j}^{a_l})\|.$$

(b) Für alle Knoten (h, i) zur Zeit R_j : ($h = 1, \dots, m_{j-1}, i = 1, \dots, n_{R_j}(h)$)

i. Bestimme ein $a_{h'} \in A$ so dass Szenario $\zeta^{a_{h'}}$ nahe an Knoten (h, i) liegt:

$$a_{h'} \in \arg \min_{a_l \in A} \sum_{i' \in C_{R_j}^{(h,i)}} \|(\zeta_{R_j-\tau}^{i'}, \dots, \zeta_{R_j}^{i'}) - (\zeta_{R_j-\tau}^{a_l}, \dots, \zeta_{R_j}^{a_l})\|.$$

ii. Knoten (h, i) zur Zeit R_j erhält Teilbaum h' für den Zeitraum $[R_j + 1, R_{j+1}]$; die Ausgangs-Szenarien die durch Knoten (h, i) gehen, werden zur Konstruktion des Teilbaums h' genutzt. Setze dazu

$$\underline{C}_{R_j}^{(h')} := \underline{C}_{R_j}^{(h')} \cup C_{R_j}^{(h,i)}.$$

2. Für $h = 1, \dots, m_j$: (**Konstruktion der Teilbäume für** $[R_j + 1, R_{j+1}]$)

(a) Finde eine Menge $A = \{a_1, \dots, a_{n_{R_j+1}(h)}\} \subset \underline{C}_{R_j}^{(h)}$ so dass der folgende Term klein wird:

$$\sum_{i' \in \underline{C}_{R_j}^{(h)}} \min_{a_l \in A} \|\zeta_{R_j+1}^{i'} - \zeta_{R_j+1}^{a_l}\|.$$

Für alle $i' \in \underline{C}_{R_j}^{(h)}$: Wähle $a_i \in \arg \min_{a_l \in A} \|\zeta_{R_j+1}^{i'} - \zeta_{R_j+1}^{a_l}\|$ und füge i' zu $C_{R_j+1}^{(h,i)}$ hinzu.

Definiere den Wert von ξ_{R_j+1} auf dem Knoten (h, i) durch $\xi_{R_j+1}^{(h,i)} := \zeta_{R_j+1}^{a_i}$.

Für alle Knoten (\tilde{h}, \tilde{i}) zur Zeit R_j ($\tilde{h} = 1, \dots, m_{j-1}, \tilde{i} = 1, \dots, n_{R_j}(\tilde{h})$) mit $C_{R_j}^{(\tilde{h}, \tilde{i})} \subset \underline{C}_{R_j}^{(h)}$: definiere die Übergangswahrscheinlichkeit von (\tilde{h}, \tilde{i}) zu (h, i) :

$$\mathbb{P}[\xi_{R_j+1} = \xi_{R_j+1}^{(h,i)} \mid \xi_{[R_j]} = \xi_{[R_j]}^{(\tilde{h}, \tilde{i})}] := |C_{R_j+1}^{(h,i)}| / |C_{R_j}^{(\tilde{h}, \tilde{i})}|.$$

(b) Für $t = R_j + 1, \dots, R_{j+1} - 1$:

Für alle Knoten \tilde{i} des Teilbaumes h zur Zeit t :

Finde eine Menge $A = \{a_1, \dots, a_{s_{t+1}(\tilde{i})}\} \subset C_t^{(h, \tilde{i})}$ so dass der folgende Term klein wird:

$$\sum_{i' \in C_t^{(h, \tilde{i})}} \min_{a_l \in A} \|\zeta_{t+1}^{i'} - \zeta_{t+1}^{a_l}\|.$$

Für alle $i' \in C_t^{(h, \tilde{i})}$: Wähle $a_i \in \arg \min_{a_l \in A} \|\zeta_{t+1}^{i'} - \zeta_{t+1}^{a_l}\|$ und füge i' zu $C_{t+1}^{(h, \tilde{i})}$ hinzu.

Setze $\xi_{t+1}^{(h, \tilde{i})} := \zeta_{t+1}^{a_i}$ und $\mathbb{P}[\xi_{t+1} = \xi_{t+1}^{(h, \tilde{i})} \mid \xi_{[t]} = \xi_{[t]}^{(\tilde{h}, \tilde{i})}] := |C_{t+1}^{(h, \tilde{i})}| / |C_t^{(\tilde{h}, \tilde{i})}|.$

Abbildung 10.8: Konstruktion eines rekombinierenden Szenariobaums.

$\zeta_t^{i'}$	Wert von Ausgangs-Szenario i' zur Zeit t
$\xi_t^{(i)}$	Wert der Zufallsvariablen ξ_t in Knoten i
m_j	Anzahl der verschiedenen Teilbäume die zur Zeit R_j beginnen ($m_0 = 1$)
(h, i)	Knoten i des Teilbaumes h
$n_t(h)$	Anzahl der Knoten des Teilbaumes h zur Zeit R_j
$s_{t+1}(i)$	Anzahl der Knoten zur Zeit $t + 1$, die von Knoten i zur Zeit t ausgehen
τ	Länge des zurückschauenden Zeithorizonts beim Rekombinieren
$C_t^{(h,i)}$	Teilmenge von $\{1, \dots, N\}$, bezeichnet die Ausgangs-Szenarien $\zeta^{i'}$ die zur Zeit t im Knoten (h, i) liegen
$\underline{C}_{R_j}^{(h)}$	Teilmenge von $\{1, \dots, N\}$, bezeichnet die Ausgangs-Szenarien $\zeta^{i'}$ die in Teilbaum h mit Beginn zur Zeit R_j liegen

Tabelle 10.1: Bezeichnungen innerhalb des Baumkonstruktionsalgorithmus.

10.5.1 Stabilität

Mit dem oben vorgestellten Approximationsverfahren kann man aus einer Menge von Szenarien ζ eines zugrunde liegenden stochastischen (Original-)Prozesses ζ einen rekombinierenden Szenariobaum konstruieren. Es stellt sich die Frage, unter welchen Bedingungen diese Art der Approximation *konsistent* ist, d.h. wann beispielsweise der mit Hilfe eines solchen Szenariobaums ξ berechnete Optimalwert $v(\xi)$ nahe am Optimalwert $v(\zeta)$ des Originalproblems liegt. In [14] wurde gezeigt, dass unter gewissen Voraussetzungen an den Ausgangsprozess ζ sowie die Struktur des Optimierungsproblems der Optimalwert $v(\xi)$ des approximierten Problems bei Verfeinerung der Diskretisierung in der Tat gegen den Optimalwert $v(\zeta)$ des Originalproblems konvergiert. Neben gewissen technischen Voraussetzungen ist dies die folgende Voraussetzung an die Struktur des Originalprozesses ζ :

Annahme. Der Prozess ζ ist durch die folgende Rekursion definiert:

$$\zeta_{t+1} := g_{t+1}((\zeta_{t-\tau}, \dots, \zeta_t), \varepsilon_{t+1}), \quad (10.6)$$

wobei $(\varepsilon_t)_{t=1, \dots, T}$ eine Folge unabhängiger Zufallsvariablen ist und g_{t+1} eine Abbildung von $\mathbb{R}^{s \cdot (\tau+1)}$ nach \mathbb{R}^s , die die folgende Lipschitz-Bedingung erfüllt: Es gibt eine Konstante $L \geq 0$, so dass

$$\left\| g_{t+1}((\zeta_{t-\tau}, \dots, \zeta_t), \varepsilon_{t+1}) - g_{t+1}((\hat{\zeta}_{t-\tau}, \dots, \hat{\zeta}_t), \varepsilon_{t+1}) \right\| \leq L \sum_{r=t-\tau}^t \|\zeta_r - \hat{\zeta}_r\| \quad (10.7)$$

für alle möglichen Werte $(\zeta_{t-\tau}, \dots, \zeta_t), (\hat{\zeta}_{t-\tau}, \dots, \hat{\zeta}_t)$ des Zufallsvektors $(\zeta_{t-\tau}, \dots, \zeta_t)$

und alle Werte ε_{t+1} von ε_{t+1} . Bedingung (10.7) kann relaxiert werden, indem man einen Lipschitz Faktor L betrachtet, der von den Werten ζ , $\hat{\zeta}$ und ε_{t+1} abhängen darf.

Diese Bedingungen werden beispielsweise durch die in einer Vielzahl von Anwendungen genutzten autoregressiven Zeitreihenmodelle erfüllt.

10.6 Schlussbemerkungen

In diesem Kapitel wurde eine Modifikation der Nested Benders Dekomposition für mehrstufige stochastische Optimierungsprobleme vorgestellt, die auf dem Konzept rekombinierender Szenariobäume beruht. Der Dekompositionsansatz ermöglicht dabei das "dynamische Rekombinieren" von Teilbäumen in Abhängigkeit von den betrachteten Werten der Entscheidungsvariablen. Dies erlaubt es, zeitkoppelnde Restriktionen in das Konzept rekombinierender Szenariobäume zu integrieren. Weiterhin wurde ein auf [12] basierender Algorithmus zur Konstruktion von rekombinierenden Szenariobäumen vorgestellt.

Wie in [7] gezeigt, lässt sich das vorgestellte Dekompositionsverfahren unmittelbar auf Probleme mit diskreten Entscheidungen in der *ersten* Stufe verallgemeinern. Diskrete Entscheidungen *nach dem ersten* Rekombinierungszeitpunkt führen dagegen im Allgemeinen zum Verlust der Konvexität der zu approximierenden Kostenfunktionen, so dass sich eine wie in Abschnitt 10.2.3 diskutierte Schnittebenenapproximation nicht ohne Weiteres bestimmen lässt. Eine Möglichkeit zur Berücksichtigung diskreter Entscheidungsvariablen besteht möglicherweise in einer Verallgemeinerung des in [20] vorgestellten Dekompositionsverfahren für zweistufige gemischt-binäre stochastische Optimierungsprobleme auf den mehrstufigen Fall.

Eine weitere mögliche Verallgemeinerung besteht in der Berücksichtigung von *Risikomaßen* in der Zielfunktion, siehe [5, 6, 17] sowie Kapitel 12. Wie in [6] gezeigt, erlaubt die Verwendung mehrperiodischer polyhedraler Risikomaße wie dem Average-Value-at-Risk weiterhin die Anwendung verschiedener Dekompositionsverfahren, u.a. auch der in diesem Kapitel verwendeten zeitlichen Dekomposition.

Danksagung. Wir danken dem BMBF für die finanzielle Unterstützung. Weiterhin danken wir Prof. Werner Römisch für seine stetige Unterstützung und Dr. Holger Heitsch für hilfreiche Diskussionen.

10.7 Literatur

- [1] BIRGE, J.R.: Decomposition and Partitioning Methods for Multistage Stochastic Programming. In: *Operations Research* 33 (1985), Nr. 5, S. 989–1007

- [2] COX, J. ; ROSS, S. ; RUBINSTEIN, M.: Option Pricing: a simplified approach. In: *Journal on Financial Economics* 7 (1979), S. 229–263
- [3] DUPAČOVÁ, J. ; CONSIGLI, G. ; WALLACE, S.W.: Scenarios for multistage stochastic programming. In: *Annals of Operations Research* 100 (2000), S. 25–53
- [4] DUPAČOVÁ, J. ; GRÖWE-KUSKA, N. ; RÖMISCH, W.: Scenarios reduction in stochastic programming: An approach using probability metrics. In: *Mathematical Programming* 95(A) (2003), S. 493–511
- [5] EICHHORN, A.: *Stochastic Programming Recourse Models: Approximation, Risk Aversion, Applications in Energy*, Humboldt-Universität zu Berlin, Diss., 2007. – Logos Verlag (Berlin)
- [6] EICHHORN, A. ; RÖMISCH, W.: Polyhedral risk measures in stochastic programming. In: *SIAM Journal on Optimization* 16 (2005), S. 69–95
- [7] EPE, A. ; KÜCHLER, C. ; RÖMISCH, W. ; VIGERSKE, S. ; WAGNER, H.-J. ; WEBER, C. ; WOLL, O.: Stochastische Optimierung mit rekombinierenden Szenariobäumen – Analyse dezentraler Energieversorgung mit Windenergie und Speichern. In: *Optimierung in der Energiewirtschaft*. Düsseldorf : VDI-Verlag, 2007 (VDI-Berichte 2018), S. 3–13
- [8] FELIX, B. ; WEBER, C.: Bewertung von unterschiedlichen Gasspeicherkonfigurationen mittels Verwendung von rekombinierenden Bäumen. In: *Optimierung in der Energiewirtschaft*. Düsseldorf : VDI-Verlag, 2007 (VDI-Berichte 2018), S. 107–120
- [9] GASSMANN, H.I.: MSLiP: a computer code for the multistage stochastic linear programming problem. In: *Mathematical Programming* 47 (1990), S. 407–423
- [10] GRÖWE-KUSKA, N. ; HEITSCH, H. ; RÖMISCH, W.: Scenario Reduction and Scenario Tree Construction for Power Management Problems. In: BORGHETTI, A. (Hrsg.) ; NUCCI, C.A. (Hrsg.) ; PAOLONE, M. (Hrsg.): *IEEE Bologna Power Tech Proceedings*, 2003
- [11] HEITSCH, H. ; RÖMISCH, W.: Scenario Reduction Algorithms in Stochastic Programming. In: *Computational Optimization and Applications* 24 (2003), S. 187–206
- [12] HEITSCH, H. ; RÖMISCH, W.: Scenario tree modeling for multistage stochastic programs. In: *Mathematical Programming* (2008). – noch nicht erschienen

- [13] INFANGER, G. ; MORTON, D.P.: Cut sharing for multistage stochastic linear programs with interstage dependency. In: *Mathematical Programming* 75 (1996), S. 241–256
- [14] KÜCHLER, C.: *Stability, Approximation, and Decomposition in Two- and Multistage Stochastic Programming*, Humboldt-Universität zu Berlin, Diss., 2008. – in Vorbereitung
- [15] KÜCHLER, C. ; VIGERSKE, S.: Decomposition of Multistage Stochastic Programs with Recombining Scenario Trees. In: *Stochastic Programming E-Print Series* 9 (2007). – <http://www.speps.org>
- [16] PEREIRA, M.V.F. ; PINTO, L.M.V.G.: Multi-stage stochastic optimization applied to energy planning. In: *Mathematical Programming* 52 (1991), S. 359–375
- [17] PFLUG, G.Ch. ; RÖMISCH, W.: *Modeling, Measuring and Managing Risk*. Singapore : World Scientific, 2007
- [18] RUSZCZYŃSKI, A.: Decomposition Methods. In: [19], Kapitel 3, S. 141–221
- [19] RUSZCZYŃSKI, A. (Hrsg.) ; SHAPIRO, A. (Hrsg.): *Handbooks in Operations Research and Management Science*. Bd. 10: *Stochastic Programming*. Amsterdam : Elsevier, 2003
- [20] SEN, S. ; HIGLE, J.L.: The C^3 theorem and a D^2 algorithm for large scale stochastic mixed-integer programming: Set convexification. In: *Mathematical Programming* 104 (2005), Nr. 1, S. 1–20
- [21] SWIDER, D. ; WEBER, C.: The costs of wind's intermittency in Germany: application of a stochastic electricity market model. In: *European Transactions on Electrical Power* 17 (2007), S. 151–172
- [22] VAN SLYKE, R.M. ; WETS, R.: L-Shaped Linear Programs with Applications to Optimal Control and Stochastic Programming. In: *SIAM Journal of Applied Mathematics* 17 (1969), Nr. 4, S. 638–663