

## 6 Das Branch-and-Cut Verfahren

Bei dem im Folgenden vorgestellten Branch-and-Cut Ansatz werden einige Cuts des Supernode processings nicht nur am Ausgangsknoten, sondern an allen oder an ausgewählten Knoten innerhalb des Enumerationsbaums angewandt. Die Idee, Cuts innerhalb des Baumes abzuleiten, geht zurück auf [PaRi91], die den Branch-and-Cut Ansatz zur Lösung großer Traveling Salesman Probleme nutzten. In den folgenden Jahren gab es eine Reihe weiterer Veröffentlichungen dazu zählen u. a. [AnCF03; CMLW97; JoNS00; Padb01; Mitc99; JüRT95]. Eine der Hauptschwierigkeiten mit der sich viele Veröffentlichungen jedoch nur am Rande beschäftigen liegt darin, eine allgemeingültige Architektur zu entwickeln, die einen sinnvollen Branch-and-Cut Einsatz bei unterschiedlichsten Modellen ermöglicht. Obwohl dem Branch-and-Cut Ansatz theoretisch große Aufmerksamkeit geschenkt wurde, wird er in den bedeutenden Optimierungssystemen, wenn überhaupt nur ansatzweise eingesetzt. Diese Tatsache legt die Vermutung nahe, dass bei der tatsächlichen Anwendung des Branch-and-Cut Verfahrens noch Verbesserungspotenzial vorhanden ist.

### 6.1 Grundstruktur eines Branch-and-Cut Ansatzes

Im Prinzip können ganzzahlige Optimierungsprobleme sowohl mit einem reinen Branch-and-Bound Verfahren als auch mit reinen Schnittebenen-Verfahren gelöst werden. Je komplexer die gemischt-ganzzahligen Probleme jedoch werden, desto problematischer wird es, sie ohne eine Kombination der Verfahren zu lösen. Für eine solche Kombination spricht auch die Tatsache, dass nur mittels Schnittebenen, die für eine vollständige polyedrische Beschreibung der konvexen Hülle der Lösungsmenge ausreichen, die optimale IP-Lösung gefunden werden kann. Andere Schnittebenen können möglicherweise nicht mehr generiert werden, obwohl die optimale IP-Lösung noch nicht gefunden wurde. Daher ist es allgemein üblich, die beiden Herangehensweisen zu kombinieren.

Die Verbindung der beiden Verfahren sieht traditionell so aus, dass im ersten Schritt von dem Originalmodell Schnittebenen abgeleitet werden. Darauf aufbauend erfolgt im zweiten Schritt mit dem Branch-and-Bound Verfahren die Identifizierung der optimalen ganzzahligen Lösung.

Der Branch-and-Cut Ansatz, der im Folgenden näher betrachtet werden soll, sieht vor, dass die Schnittebenen nicht nur am Ausgangsknoten des Branch-and-Bound Verfahrens abgeleitet werden, sondern auch innerhalb des Baumes an allen oder zumindest ausgewählten Knoten.

Der Ablauf des Branch-and-Cut Verfahrens wird in Abbildung 6.1 veranschaulicht.

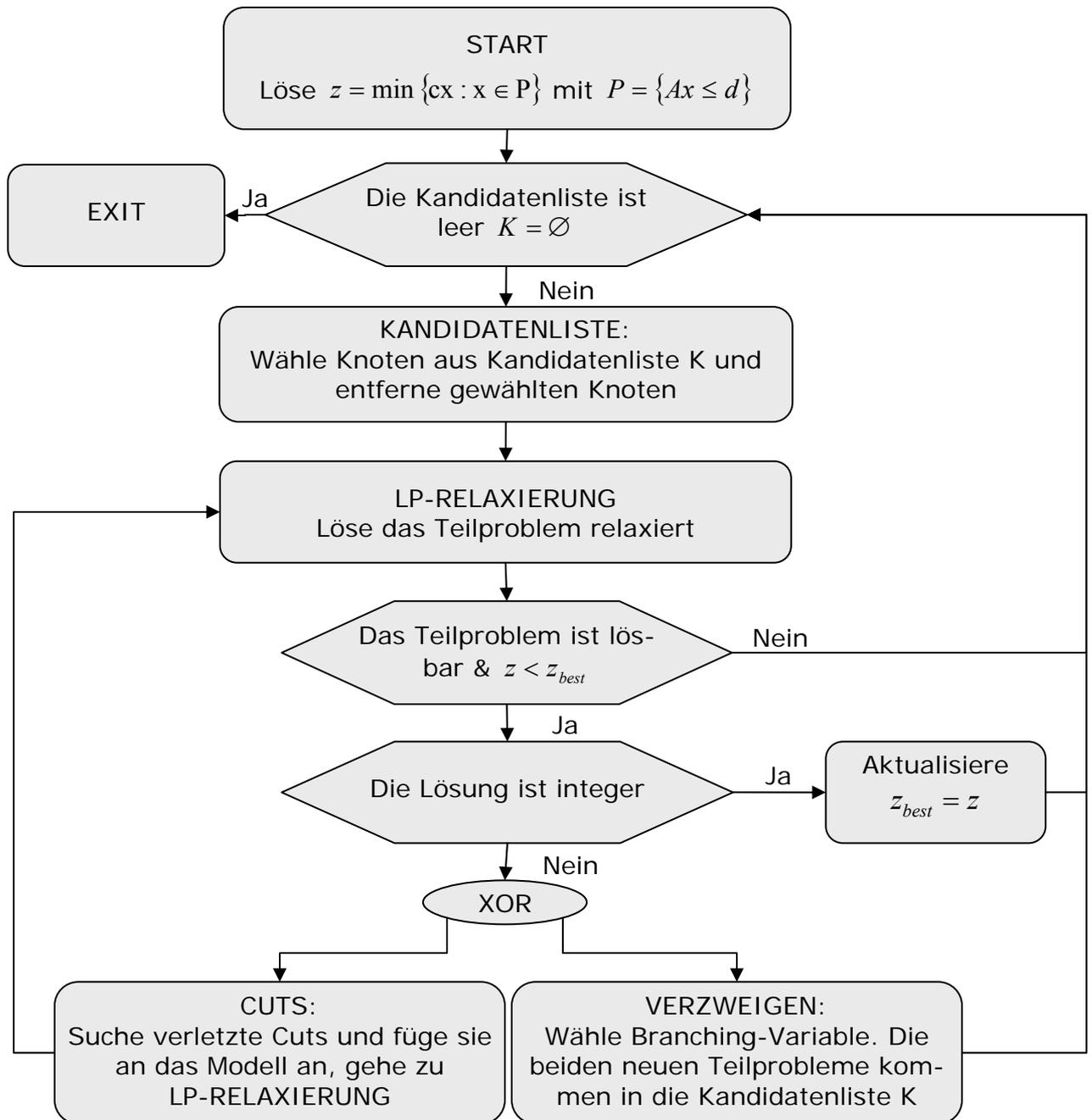


Abbildung 6.1: Das Branch-and-Cut Verfahren

Als erstes muss das Problem relaxiert gelöst werden, was den Ausgangspunkt für das Schnittebenen- und das Branch-and-Bound Verfahren darstellt. Hier können bei Bedarf problemlos

sämtliche Techniken des Supernode processings angewandt werden. Das Modell wird somit modifiziert, *bevor* das klassische Branch-and-Bound Verfahren zum Einsatz kommt. Nachdem die Techniken des Supernode processings erschöpft sind, wird, wie in Kapitel 5.1 beschrieben, mit dem Enumerationsbaum begonnen. Der entscheidende Unterschied zwischen Branch-and-Bound und Branch-and-Cut Verfahren tritt auf, wenn ein Teilproblem nicht vollständig untersucht werden konnte. Das Branch-and-Bound Verfahren wählt in diesem Fall eine Branching-Variable aus und das Teilproblem wird weiter verzweigt. Beim Branch-and-Cut Ansatz ist das nur eine von zwei Möglichkeiten. Neben dem Verzweigen besteht die Möglichkeit, nach Schnittebenen zu suchen. Bestenfalls führt das Hinzufügen von Schnittebenen nach erneutem Lösen der LP-Relaxierung dann dazu, dass ein Teilproblem vollständig untersucht ist und nicht weiter verzweigt werden muss. Ansonsten wird zumindest der Lösungsraum an diesem und ggf. auch an anderen Knoten durch die Cuts weiter eingeschränkt. Diese Entscheidung, an einem Knoten weiter zu verzweigen oder nach Cuts zu suchen, ist von fundamentaler Bedeutung im Rahmen des Branch-and-Cut Ansatzes.

## 6.2 Entscheidung zwischen Verzweigen oder Anfügen einer Schnittebene

Der Lösungsraum kann innerhalb des Baumes durch global gültige Cuts eingeschränkt werden. Es eröffnet sich für jeden einzelnen Knoten eine Alternative. Es besteht die Möglichkeit, nach verletzten Cuts zu suchen. Alternativ kann nach Auswahl einer Branching-Variablen weiter verzweigt werden. Auf diese Entscheidung soll im folgenden Abschnitt näher eingegangen werden. Dazu werden Strategien vorgestellt, die festlegen, an welchen Knoten nach verletzten Cuts gesucht wird und an welchen nicht.

### 6.2.1 Strategien zur Ermittlung eines ausgewählten Knotens

Es gibt drei einfache Begründungen, warum nicht an jedem Knoten nach Cuts gesucht werden sollte:

- In einem Modell werden zu viele Cuts gefunden. Der durch die Einschränkung des Lösungsraums entstehende Vorteil ist geringer als der Nachteil, der mit der Vergrößerung des Modells und dem während der Suche anfallenden Aufwand einhergeht.
- In einem Modell werden zu wenig Cuts bzw. gar keine gefunden. Bei einigen Klassen von Cuts erübrigt sich unter Umständen die Frage, ob ein Cut an das Modell angehängen werden soll, weil keiner gefunden werden kann. Um allerdings zu dem Ergebnis zu kommen, dass es keine verletzten Cuts gibt, muss danach gesucht worden sein. Diese Suche kostet Zeit und sollte nicht an jedem Knoten des Baumes durchgeführt werden.
- Das Einfügen von Cuts erfordert darüber hinaus eine Reorganisation der Datenstrukturen für die LP-Optimierung. Diese Zeit kann mehr ins Gewicht fallen, als die Reduktion der Anzahl zu lösender Teilprobleme.

Es stellt sich also die Frage, wie die Anzahl der ausgewählten Knoten, an denen nach Cuts gesucht wird, beschränkt werden soll. Dafür sind mehrere Strategien denkbar.

Im Rahmen dieser Arbeit wurden mehrere Modelle auf bestimmte Kriterien hin untersucht. Die verschiedenen Größen, von denen angenommen wurde, dass sie u. U. zu einer sinnvollen Strategie führen könnten, sollen zunächst einleitend vorgestellt werden:

- Der Prozentuale Anteil an ganzzahligen Variablen in einem Modell.
- Die während des Supernode processing erzielte relative Verbesserung (gemessen an

$$\text{der Veränderung des Zielfunktionswertes) } IIP = \frac{|Z_{LP} - Z_{PP}|}{1 + |Z_{LP}|} * 100$$

wobei  $Z_{LP}$  der Zielfunktionswert der Lösung des LPs ist und  $Z_{PP}$  der Zielfunktionswert nach dem Supernode processing.

- Globaler relativer Gap:  $GRG = \frac{|UB_g - LB_g|}{1 + |LB_g|} * 100$

- Relativer Gap für Knoten  $p$ :  $KRG = \frac{|UB_g - Z_p|}{1 + |Z_p|} * 100$

- Globaler absoluter Gap:  $GAG = |LB_g - UB_g|$

- Entfernung zur globalen Untergrenze des Knotens  $p$ :  $KD = |LB_g - Z_p|$

wobei  $LB_g$  und  $UB_g$  für die aktuelle globale Unter- bzw. Obergrenze steht und  $Z_p$  für den Zielfunktionswert des Teilproblems am Knoten  $p$ . Wurde noch keine globale Obergrenze gefunden, sind mehrere Vorgehensweisen denkbar. Das Kriterium kann z.B. erst angewandt werden, wenn eine globale Obergrenze gefunden ist, eine globale Obergrenze kann geschätzt werden oder es wird zu einer der anderen Strategien gewechselt, solange noch keine globale Obergrenze gefunden ist.

Mit Hilfe dieser und weiterer Größen wurden in dieser Arbeit verschiedene Strategien aufgestellt, die dafür sorgen, dass nur an einzelnen ausgewählten Knoten nach Cuts gesucht wird. Diese Strategien sollen im Folgenden vorgestellt und diskutiert werden.

Die Suche erfolgt:

- Nur am ersten Knoten:  
Diese Strategie entspricht dem klassischen Branch-and-Bound, dem im Rahmen des Supernode processings die Cutgenerierung vorgeschaltet ist.

- An jedem Knoten:  
Es wird an jedem Knoten nach Cuts gesucht, bis entweder kein Platz mehr für weitere Cuts vorhanden ist oder  $k$  Knoten hintereinander keine Cuts gefunden wurden.
- An jedem  $k$ -ten Knoten:  
Bei dieser Strategie ist der Wert für  $k$  besonders ausschlaggebend. Es kann davon ausgegangen werden, dass rein zufällig ein Knoten ausgewählt wird, unabhängig sowohl von seiner Beschaffenheit als auch davon, an welcher Stelle im Baum er sich befindet.

Die folgenden Strategien sollen mit in Betracht ziehen, wie weit die Suche nach der optimalen ganzzahligen Lösung vorangeschritten ist.

- An den ersten  $k$  Knoten:  
Die Anzahl der Knoten, an denen nach Cuts gesucht wird, ist bei dieser Art der Einschränkung unabhängig von der Größe des Baumes. Es handelt sich dabei um eine sehr *statische* Strategie. Ob diese Art der Einschränkung überhaupt sinnvoll ist, hängt sehr davon ab, welche Knotenauswahl Strategie verwendet wird.
- Wenn die relaxierte Lösung mehr als  $k$  Prozent vom globalen relativen Gap entfernt ist:  
Ist dieser Wert noch sehr groß, erscheint es sinnvoll den Lösungsraum mittels zusätzlicher Schnittebenen einzugrenzen.
- Auf den ersten  $k$  Ebenen des Baumes:  
Bei dieser Strategie ist die Anzahl der abgeleiteten Cuts unabhängig von der Länge der Suche während des Branch-and-Cut Prozesses.

Die bisher beschriebenen Verfahren haben zum Nachteil, dass die Eigenschaften eines Knotens bei der Entscheidung, ob nach Cuts gesucht wird oder nicht, keine Beachtung finden. Die folgenden Strategien beziehen sich daher auf verschiedene Knoteneigenschaften.

- Wenn der relative Gap des aktuellen Knotens (KRG) größer  $k$  ist:  
Bei dieser Strategie wird nur an den Knoten nach Cuts gesucht, die noch sehr weit von einer globalen Obergrenze entfernt sind.
- Wenn die Anzahl der fraktionellen integer Variablen  $k\%$  der Anzahl der fraktionellen Variablen am ersten Knoten ausmacht:  
Insbesondere, wenn in der Lösung der LP-Relaxierung nur noch wenige ganzzahlige Variablen fraktionell sind, kann eine Cutgenerierung zur unnötigen Verlängerung der

Lösungszeit führen. Dagegen spricht allerdings das Argument, dass die Wahrscheinlichkeit der Verletzung eines Cuts umso größer ist, je mehr Variablen fixiert sind.

- Wenn die Differenz zwischen dem relativen Gap des aktuellen Knotens und dem globalen relativen Gap kleiner  $k$  ist:

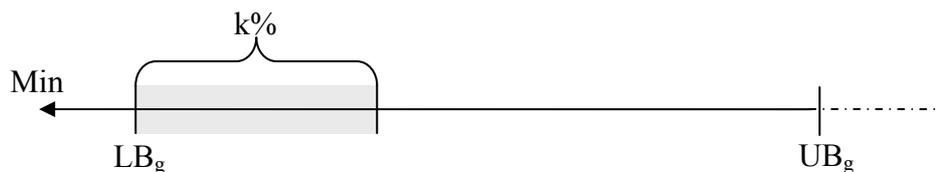
Wenn der relative Gap für einen Knoten nahe an dem globalen relativen Gap ist, heißt das, dass sich der aktuelle Knoten in der Nähe der globalen Untergrenze befinden muss.

$$GRG - KRG \leq k$$

- Wenn die Differenz zwischen dem Zielfunktionswert eines Knotens und der globalen Untergrenze kleiner als  $k\%$  des aktuellen globalen absoluten Gaps ist:

Hier soll die Nähe zur globalen Untergrenze ausgedrückt werden.

$$KD \leq \frac{k}{100} * GAG$$



Mit Hilfe dieses Wertes können alle Knoten identifiziert werden, deren Zielfunktionswert in den markierten Bereich fällt.

Bei Anwendung der beiden letzten Strategien wird ein besonderer Schwerpunkt darauf gelegt, möglichst schnell die globale Untergrenze anzuheben. Im Rahmen der lokalen Gomory Cuts (Kapitel 6.6.2.3) finden sie besondere Bedeutung.

## 6.2.2 Faktoren, die die Strategien beeinflussen

Bevor eine oder mehrere der beschriebenen Strategien zum Einsatz kommen kann, muss ein möglichst guter Wert für  $k$  gefunden werden. Die im Folgenden vorgestellten Faktoren sollen einen Einfluss auf den Wert von  $k$  haben.

- Fortschritt während des Supernode processings:  
Wurde schon viel Fortschritt im Supernode processing erzielt, spricht das einerseits dafür, dass das Modell Potenzial für Cuts bietet. Andererseits bedeutet ein hoher Fortschritt aber auch, dass auch schon viel von dem Potenzial genutzt wurde. Dies gilt ins-

besondere für Implication und Clique Cuts. Davon abgesehen kann der Fortschritt auch von den anderen Techniken, die auch während des Supernode processing angewandt werden herrühren.

- Der prozentuale Anteil an ganzzahligen Variablen in einem Modell:  
Je weniger Variablen in einem Modell ganzzahlig sind, desto geringer ist die Wahrscheinlichkeit, Cuts zu finden.
- Die Cutsuche ist im Verhältnis zu zeitaufwendig:  
Wenn die Zeit, die benötigt wird um ein LP zu lösen, deutlich geringer ist als die Zeit, die während der Cutsuche verbraucht wird, sollte weniger oft nach Cuts gesucht werden.

Natürlich kann der Parameter  $k$  auch im Laufe des Branch-and-Cut Prozesses angepasst werden:

- Erfolglose Suche:  
Insofern die Suche nach einer bestimmten Art von Cuts wiederholt zu keinem Erfolg führte, sollte die Suche nach dieser Art von Cuts eingeschränkt werden.
- Fortschritt durch Cuts:  
Die Anzahl der Cuts wird mit der durch sie verursachten Verbesserung des Zielfunktionswertes in Beziehung gebracht.
- Fortschritt durch Cuts im Verhältnis zum Fortschritt durch Verzweigen:  
Nach einer bestimmten Anzahl von Knoten, wird die Verbesserung des Zielfunktionswertes durch das Anfügen von Cuts verglichen mit der Verbesserung durch das Anfügen einer Verzweigungsrestriktion.

Im Rahmen dieser Arbeit konnte festgestellt werden, dass, von wenigen Ausnahmen abgesehen, das Ableiten von Cuts innerhalb des Baumes immer dazu führt, dass weniger Knoten abgearbeitet werden müssen.

### 6.2.3 Löschen von nicht bindenden Schnittebenen

Je mehr Cuts an das Modell angehängt werden, desto länger dauert die einzelne LP-Optimierung, die an jedem Knoten durchgeführt werden muss. Um das Modell möglichst klein zu halten, können *nicht bindende* Cuts wieder aus dem Modell herausgenommen wer-

den. Zur Feststellung, wann ein Cut nicht bindend ist, muss noch einmal die Struktur eines Modells genauer betrachtet werden.

Es werden für jede  $\geq$ -Restriktion und jede  $\leq$ -Restriktion logische Variablen eingeführt. Die Restriktionen  $\sum_j x_j \leq b$  und  $\sum_j x_j \geq b$  werden mit Hilfe von logischen Variablen folgendermaßen dargestellt:

$$\sum_j x_j \leq b \Rightarrow \sum_j x_j + x_l = b$$

$$\sum_j x_j \geq b \Rightarrow \sum_j x_j - x_u = b$$

wobei  $x_u, x_l \geq 0$ .

Wenn eine Restriktion bindend ist, ist  $\sum_j x_j = b$  und somit  $x_u, x_l = 0$ . Eine nicht bindende

Restriktion stellt somit eine Beschränkung dar, die von der optimalen Lösung nicht voll ausgenutzt wird. Würde die Beschränkung „gelockert“ werden, würde damit keine Verbesserung der optimalen Lösung einhergehen. Nicht bindende Restriktionen können also über ihre logischen Variablen identifiziert werden. Ist die logische Variable eines Cuts  $> 0$ , ist dieser nicht bindend.

Da der Aufwand, einen Cut wieder aus dem Modell zu entfernen, nicht unerheblich ist, sollte auch hier einer bestimmten Strategie folgend vorgegangen werden. Ein Cut wird aus dem Modell entfernt, wenn er  $k$ -mal hintereinander nicht bindend war. Der Wert für  $k$  sollte dabei von den Eigenschaften des Cuts abhängig sein. Ein nicht bindender Cut, der die LP-Optimierung sehr verlangsamt, sollte möglichst schnell wieder inaktiviert werden. Somit stellt sich das Problem, ein Kriterium zu finden, welches einen Anhaltspunkt dafür gibt, welcher Cut im Vergleich zu einem anderen eher zur Verlangsamung der LP-Optimierung beiträgt. Im Rahmen dieser Arbeit wurde das abgewandelte Markowitz Kriterium, welches detailliert in Kapitel 4.7.5.1 vorgestellt wird, verwandt. Es wird die durchschnittliche Anzahl an Nicht-Null-Elementen ermittelt, die die Spalten der Variablen, die Teil des Cuts sind, haben.

$$MK = \sum_{j \in C} (n_j - 1) / |C|$$

wobei  $n_j$  die Anzahl an Nicht-Null-Elemente der Spalte  $j$  und  $C$  die Menge der Nicht-Null-Elemente des Cuts darstellt. Je höher diese Zahl ist, desto wahrscheinlicher ist es, dass dieser

Cut während der LP-Optimierung für die Erzeugung von vielen Nicht-Null-Elementen sorgt und somit möglichst schnell wieder inaktiviert werden sollte.

Eine weitere, entscheidende Strategie liegt darin, Cuts nur dann zu überprüfen, wenn eine bestimmte Anzahl von Nicht-Null-Elementen dem Modell neu hinzugefügt worden ist. Die erforderliche Obergrenze wird auf Basis der Anzahl der Nicht-Null-Elemente des Modells errechnet.

Durch das Löschen von Cuts kann der Fall eintreten, dass an einem Knoten die LP-Optimierung einen kleineren Wert ergibt als die globale Untergrenze. Ohne das Inaktivieren von Cuts konnte dieser Fall nicht eintreten, da das Anfügen von Schnittebenen immer zu einer Verschärfung führt. Um die bereits gewonnene Reduktion des Gaps nicht wieder zu verlieren, sollte in diesem Fall die Zielfunktion als Restriktion mit der globalen Untergrenze als Untergrenze an das Modell angefügt werden. Das gleiche Prinzip findet auch Anwendung bei der Verwendung von lokal gültigen Gomory Cuts (Kapitel 6.6.2.3).

## 6.2.4 Ein Cutpool zur Aufbewahrung von nicht verletzten Schnittebenen

Eine weitere Überlegung besteht darin, einen *Cutpool* [PaRi91] anzulegen. Nicht aktive Cuts können darin aufbewahrt werden. Der Aufwand, den Cutpool nach verletzten Schnittebenen zu durchsuchen, ist weniger aufwendig als der erneute Durchlauf der Routinen, die Cuts erst neu identifizieren.

Ein Cutpool kann unterschiedliche Arten von Cuts beinhalten. Dies können zum Beispiel Cuts sein, die, nachdem sie an  $k$ -Knoten nicht bindend waren, aus dem Modell herausgenommen werden. Des Weiteren kann ein Cutpool Cuts beinhalten, die zwar generiert aber nie an das Modell angehängen wurden, da sie nicht verletzt waren.

Zu der Idee eines Cutpools ist zu bemerken, dass sowohl das Speichern von Cliques als auch das Speichern von Implikationen dem Prinzip eines Cutpools sehr ähneln. Potenzielle Cuts werden gespeichert und bei Bedarf daraufhin untersucht, ob sie von der aktuellen LP-Relaxierung verletzt werden.

## 6.3 Globale und lokale Gültigkeit von Schnittebenen im Baum

Wird ein Cut innerhalb des Enumerationsbaumes im Rahmen des Branch-and-Cut Ansatzes abgeleitet, muss besonderes Augenmerk auf dessen Gültigkeit gelegt werden. Es kann zunächst nur davon ausgegangen werden, dass er für das Teilproblem von dem er abgeleitet wurde, und für alle Nachfolgerknoten dieses Teilproblems, gültig ist. D.h. er ist nicht zwangsläufig für alle und insbesondere nicht unbedingt für die Ausgangsaufgabe gültig.

Betrachten wir zunächst folgendes 0-1-Problem:

$$\begin{aligned} \min \quad & \sum_{j \in B} c_j x_j \\ & Ax \leq d \\ & x_j \in \{0,1\}, \quad \forall j \in B \end{aligned}$$

Die Definitionen entsprechen denen aus Kapitel 3.2.

$F_u^i$  bzw.  $F_l^i$  sollen die Menge der auf 1 bzw. 0 fixierten Variablen am Knoten  $i$  sein.

Ein am Knoten  $i$  abgeleiteter *lokal* gültiger Cut ist nur für das Teilproblem am Knoten  $j$  gültig, wenn gilt  $F_u^i \subseteq F_u^j$  und  $F_l^i \subseteq F_l^j$ . Da  $F_u^i \subseteq F_u^j$  und  $F_l^i \subseteq F_l^j$  nur gelten, wenn  $j$  ein Nachfolgerknoten von  $i$  ist, ist ein lokaler Cut des Knotens  $i$  nur für das Teilproblem am Knoten  $i$  und die Teilprobleme der Nachfolgerknoten von  $i$  gültig.

Dagegen ist ein *global* gültiger Cut unabhängig von den Mengen  $F_u^i$  und  $F_l^i$  am Knoten, an dem er abgeleitet wird, für den gesamten Enumerationsbaum gültig.

Diese Definitionen lassen sich leicht auf allgemeine ganzzahlige Probleme übertragen. Ein lokal gültiger Cut ist nur an dem Knoten, von dem er abgeleitet wurde, und dessen Nachfolgern gültig. Alle Verzweigungen, die Einfluss auf die Herleitung eines Cuts gehabt haben können, bleiben an den Nachfolgerknoten gültig. Global gültige Cuts sind auch bei allgemeinen, ganzzahligen Problemen unabhängig von den Verzweigungen im gesamten Baum gültig.

Global gültige Cuts haben gegenüber lokal gültigen Cuts entscheidende Vorteile.

Der Speicheraufwand, der für lokal gültige Cuts erforderlich ist, ist entschieden größer. Ein global gültiger Cut kann jederzeit dem Modell hinzugefügt werden und auch Teil des Modells

bleiben. Wohingegen ein lokal gültiger Cut als zusätzliche Information an dem jeweiligen Knoten gespeichert werden muss, wenn er für die Nachfolgerknoten zur Verfügung stehen soll.

Von einem global gültigen Cut kann überall im Enumerationsbaum profitiert werden. Ein einmal abgeleiteter, global gültiger Cut wird Teil des Modells und kann somit auch in einem ganz anderen Zweig des Baumes für eine Einschränkung des Lösungsraumes sorgen.

Sowohl die Handhabung als auch die Effektivität eines global gültigen Cuts sind also einfacher bzw. höher.

Ob eine Klasse von Cuts global oder lokal gültig ist, hängt von ihrem Herleitungsverfahren ab. Entscheidend ist dabei, ob die Herleitung eines Cuts durch die Verzweigung oder Fixierung der Branching-Variablen beeinflusst wird. Beispielsweise die Verwendung der aktuellen Basis oder LP-Lösung an einem Knoten kann zur lokalen Gültigkeit eines Cuts führen.

## 6.4 Liften zur globalen Gültigkeit

### 6.4.1 Binäre Variablen

Wie im vorangegangenen Abschnitt dargelegt, ist die Handhabung global gültiger Cuts entschieden einfacher als die lokal gültiger Cuts. Im Folgenden wird eine Form des Liftings vorgestellt, durch die ein lokal gültiger Cut global gültig werden kann.

Zunächst soll ein binäres Problem betrachtet werden. Bei einem binären Problem wird die lokale Gültigkeit eines Cuts dadurch verursacht, dass er abgeleitet wird unter der Voraussetzung, dass bestimmte Variablen am Knoten  $i$  auf null oder eins fixiert sind. Sei  $F_l^i$  und  $F_u^i$ , die Menge der auf null bzw. eins fixierten Variablen am Knoten  $i$ . Global gültig kann dieser Cut werden, wenn alle Variablen aus  $F_l^i$  und  $F_u^i$ , also alle während des Branch-and-Cut Prozesses fixierten Variablen, geliftet werden. Ist z.B.  $x_j \in F_u^i$ , muss  $x_j$  geliftet werden. Dazu wird ein Lifting-Koeffizient  $\beta_j$  ermittelt, mit dem  $x_j$  Teil des Cuts wird. Durch das Aufnehmen von  $\beta_j x_j$  in den Cut soll sichergestellt werden, dass der Cut gültig ist für den Fall, dass  $x_j = 0$ .

Die Variablen aus  $F_l^i$  und  $F_u^i$  müssen auf unterschiedliche Art und Weise geliftet werden.

Zuerst soll die Vorgehensweise mit den Variablen aus  $F_u^i$  beschrieben werden. Sei  $C$  die

Menge der Nicht-Null-Elemente des zu liftenden Cuts. Aus dem Cut  $\sum_{c \in C} \alpha_c x_c \leq \alpha_0$  entsteht

durch das Liften der Variablen in  $F_u^i$ :

$$\sum_{c \in C} \alpha_c x_c + \sum_{j \in F_u^i} \beta_j x_j \leq \alpha_0 + \sum_{j \in F_u^i} \beta_j$$

wobei  $\beta_j$  der gesuchte Lifting-Koeffizient für die Variable  $j$  ist. Dieser Lifting-Koeffizient lässt sich folgendermaßen ermitteln:

$$\beta_j = z_j - \alpha_0$$

$$\text{wobei } z_j = \max_{c \in C} \sum \alpha_c x_c, x \in P \wedge x_j = 0$$

Mit  $P = \{x \in \{0,1\}^n : Ax \geq d\}$  wird die Menge der zulässigen Lösungen des Originalproblems bezeichnet. Da ein größerer Wert für  $\beta_j$  nur zu einer weiteren Auflockerung des Cuts führt, kann  $z_j$  auch relaxiert gelöst werden.

Eine ähnliche Vorgehensweise wurde bereits in Kapitel 4.7.3 beschrieben. Während dort Cuts verschärft werden sollten, sollen sie hier global gültig werden. Der entscheidende Unterschied dabei ist, dass der Lifting-Koeffizient  $\beta_j$  der Variablen  $j$  sowohl positiv als auch negativ sein kann. Ist  $\beta_j$  positiv wird der Cut aufgelockert. Damit ist sichergestellt, dass er gültig ist, sowohl wenn  $x_j = 1$ , als auch wenn  $x_j = 0$  gilt. Ist  $\beta_j$  negativ, wird der Cut sogar verschärft. Besteht der Anspruch lediglich darin, einen Cut global gültig zu machen, müssen Variablen mit einem negativen Lifting-Koeffizienten nicht in den Cut aufgenommen werden. Es spricht jedoch nichts dagegen, die ohnehin schon ermittelte Verschärfung des Cuts zu nutzen.

Das Liften der Variablen aus  $F_l^i$  verändert den originalen Cut zu

$$\sum_{c \in C} \alpha_c x_c + \sum_{j \in F_l^i} \lambda_j x_j \leq \alpha_0.$$

Der gesuchte Lifting-Koeffizient ergibt sich dabei aus

$$\lambda_j = \alpha_0 - z_j$$

$$\text{wobei } z_j = \max \sum_{c \in C} \alpha_c x_c, x \in P \wedge x_j = 1$$

$$\text{und } P = \{x \in \{0,1\} : Ax \leq d\}$$

Hier sorgen die negativen Lifting-Koeffizienten dafür, dass der Cut global gültig wird, während die positiven Lifting-Koeffizienten den Cut verschärfen. Da folglich wieder ein größerer Wert für  $z_j$  den Cut weiter auflockert, sollte auch hier der Wert für  $z_j$  durch die Lösung des relaxierten Problems erfolgen.

Insgesamt sieht ein global gültiger Cut, der am  $i$ -ten Knoten abgeleitet wurde folgendermaßen aus:

$$\sum_{c \in C} \alpha_c x_c + \sum_{j \in F_i^l} \lambda_j x_j + \sum_{j \in F_u^i} \beta_j x_j \leq \alpha_0 + \sum_{j \in F_U^i} \beta_j .$$

Je mehr Variablen in einem Modell fixiert werden, desto aufwendiger wird dieses Verfahren.

Für jede fixierte Variable muss ein Problem, gelöst werden, welches fast so komplex ist wie das Originalproblem.

## 6.4.2 Allgemeine Integervariablen

Zur Herleitung der Lifting-Koeffizienten für allgemeine Integervariablen, soll zunächst der 0-1-Fall erweitert werden. Eine Variable  $x_j$  kann entweder den Wert  $t$  oder den Wert  $s$  annehmen. Bei Ableitung des Cuts war  $x_j$  auf  $t$  fixiert. Jetzt soll betrachtet werden, welchen Wert der Lifting-Koeffizient annehmen muss damit der Cut auch gültig ist, wenn  $x_j$  den Wert  $s$  annimmt.

Ist  $t > s$  ergibt sich der Lifting-Koeffizient aus:

$$\beta_j = \frac{z_s - \alpha_0}{t - s}$$

$$\text{wobei } z_s = \max_{c \in C} \sum \alpha_c x_c, x \in P \wedge x_j = s$$

$$\text{und } P = \{x \in \mathbb{Z}_+ \mid Ax \leq d\}$$

Dieser Wert drückt die Obergrenze für den Cut aus, wenn  $x_j = s$ .

$$\text{So entsteht der geliftete Cut } \sum_{c \in C} \alpha_c x_c + \beta_j x_j \leq \alpha_0 + \beta_j t.$$

Durch Einsetzen der Werte  $t$  oder  $s$  für  $x_j$  wird deutlich, dass die Auflockerung des Cuts wirklich nur dann zum Tragen kommt, wenn  $x_j = s$ .

Ist  $s > t$  muss lediglich die Berechnung des Lifting-Koeffizienten angepasst werden.

$$\beta_j = \frac{\alpha_0 - z_s}{s - t}.$$

Diese Anpassung ist erforderlich, da ansonsten  $\beta_j$  den Cut aufgrund des falschen Vorzeichens verschärfen und nicht auflockern würde.

Handelt es sich um ein Problem mit allgemeinen Integervariablen, so werden diese nicht auf einen Wert fixiert, sondern ihre Ober- oder Untergrenze wird angepasst. Ziel ist es herauszubekommen, inwiefern sich die Obergrenze des lokal gültigen Cuts verändert, wenn die Variable  $x_j$  unterschiedliche Werte  $lb_j \leq x_j \leq ub_j$  annimmt. Dabei ist  $lb_j$  die Untergrenze und  $ub_j$  die Obergrenze der Variablen  $x_j$ . Durch den Lifting-Koeffizienten  $\beta_j$  soll sichergestellt werden, dass der Cut bei allen Werten gültig bleibt.

War  $\bar{f}_j$  die Untergrenze für  $x_j$ , als der zu liftende lokale Cut abgeleitet wurde, lässt sich der Lifting-Koeffizient  $\beta_j$  durch

$$\beta_j = \frac{z_v - \alpha_0}{\bar{f}_j - v}$$

$$\text{mit } z_v = \max_{c \in C} \sum \alpha_c x_c, x \in P \wedge x_j = v$$

$$\text{und } P = \{x \in \mathbb{Z}_+ \mid Ax \leq d\}$$

definieren, wenn  $x_j$  einen Wert  $lb_j \leq v < \bar{f}_j$  annimmt

Dieser Wertebereich wurde vorher durch die neue Untergrenze  $\bar{f}_j$  verhindert. Keiner der Werte  $lb_j \leq v < \bar{f}_j$  kann allerdings ausgeschlossen werden. Da der größte Wert für  $\beta_j$ , die maximale Veränderung des Cuts beschreibt, ist nur dieser von Interesse. Eine Untergrenze für den Lifting-Koeffizienten  $\beta_j$  lässt sich demzufolge definieren durch:

$$\max_{v=0, \dots, \bar{f}_j-1} \frac{z_v - \alpha_0}{\bar{f}_j - v} \leq \beta_j$$

Damit der Cut nach diesem Liften nicht ungültig wird für den Fall, dass  $x_j$  einen Wert aus dem Bereich  $\bar{f}_j < v \leq ub_j$  annimmt, wird eine Obergrenze für  $\beta_j$  definiert.

$$\beta_j \leq \min_{v=\bar{f}_j+1, \dots, ub_j} \frac{\alpha_0 - z_v}{v - \bar{f}_j}$$

Für diese Obergrenze ist der kleinstmögliche Wert von Interesse, da die negative Veränderung des Cuts betrachtet wird, wenn  $\bar{f}_j < v$ .

Da ohnehin sowohl eine Ober- also auch eine Untergrenze für  $\beta_j$  definiert werden, ist die Vorgehensweise dieselbe, unabhängig von der Richtung der Verzweigung.

Wird ein lokal gültiger Cut abgeleitet, muss für alle Variablen, die an einer Verzweigung beschränkt wurden, ein Koeffizient gefunden werden.

Befinden sich in der Menge  $V$  alle Variablen, die während des Branch-and-Cut Prozesses nach unten oder oben beschränkt wurden, ergibt sich der geliftete Cut:

$$\sum_{c \in C} \alpha_c x_c + \sum_{j \in V} \beta_j x_j \leq \alpha_0 + \sum_{j \in V} \beta_j \bar{f}_j$$

Für die Lifting-Koeffizienten müssen die Werte jeweils in folgendem Bereich liegen.

$$\max_{v=0, \dots, \bar{f}_j-1} \frac{z_v - \alpha_0}{\bar{f}_j - v} \leq \beta_j \leq \min_{v=\bar{f}_j+1, \dots, ub_j} \frac{\alpha_0 - z_v}{v - \bar{f}_j}$$

$$\text{wobei } z_v = \max_{c \in C} \sum \alpha_c x_c, x \in P \wedge x_j = v$$

$$\text{und } P = \{x \in \mathbb{Z}_+ \mid Ax \leq d\}$$

Im folgenden Beispiel, soll gezeigt werden, wie aus einer lokal gültigen Schnittebene eine global gültige hergeleitet werden kann. Dabei wird, wie eben beschrieben vorgegangen, um einen Lifting-Koeffizienten zu ermitteln.

*Beispiel:*

$$\begin{aligned} \min \quad & -3x_1 - 4x_2 \\ & 2x_1 + 3x_2 \leq 9 \\ & 4x_1 - 3x_2 \leq 7 \\ & x_1, x_2 \in \mathbb{Z}_+ \end{aligned}$$

*Als Verzweigungsrestriktion wird  $x_2 \leq 1$  angehängen. Dann kann die Schnittebene  $x_1 + 2x_2 \leq 4$  generiert werden. Dabei handelt es sich um eine lokal gültige Schnittebene, denn sie schneidet die optimal ganzzahlige Lösung  $x_1 = 0$ ;  $x_2 = 3$  ab. Ziel ist es, diese Schnittebenen global gültig zu machen also, auch wenn die Verzweigungsrestriktion  $x_2 \geq 2$  Teil des Modells ist.*

*Dazu muss sie erweitert werden zu  $x_1 + 2x_2 + \beta_2 x_2 \leq 4 + 1\beta_2$ .*

$$\max_{v=0} \frac{z_v - 4}{1 - v} \leq \beta_2 \leq \min_{v=2, \dots, ub_2} \frac{4 - z_v}{v - 1}$$

$$\begin{aligned} z_v = \max \quad & x_1 + 2x_2 \\ & 2x_1 + 3x_2 \leq 9 \\ & 4x_1 - 3x_2 \leq 7 \\ & x_1, x_2 \in \mathbb{Z}_+ \end{aligned}$$

*Dann ist  $z_0 = 1$ ,  $z_2 = 5$  und  $z_3 = 6$ .*

*Daraus folgt,*

$$\max\{-3\} \leq \beta_2 \leq \min\{-1; -1\} \text{ bzw. } -3 \leq \beta_2 \leq -1.$$

*Durch Einsetzen von  $\beta_2$  entstehen die global gültigen Ungleichungen:*

$$x_1 + x_2 \leq 3$$

$$x_1 - x_2 \leq 1$$

$$\text{und } x_1 \leq 2.$$

Das liften von allgemeinen integer Variablen ist ein sehr aufwendiges Verfahren, es müssen i. d. R. mehrere Maximierungsprobleme für die Ermittlung eines Lifting-Koeffizienten gelöst werden. Diese Maximierungen können auch relaxiert erfolgen, jedoch muss dann eine Auflockerung des Cuts hingenommen werden, die dazu führen kann dass dieser nicht mehr verletzt ist.

Das Liften von 0-1-Variablen ist im Prinzip nur ein Spezialfall des Liftens von allgemeinen ganzzahligen Variablen. Dadurch, dass die zu lifte Variable nur einen anderen Wert annehmen kann, kann die Ermittlung des Lifting-Koeffizienten allerdings sehr vereinfacht werden. Für weitere Literatur bezüglich des Liftens von allgemeinen Integer Variablen vgl. [CCMW98; Mitc 99].

## 6.5 Anwendung von lokal gültigen Cuts

Angenommen es wurden lokal gültige Cuts abgeleitet. Diese Cuts können durch Liften global gültig gemacht werden. Allerdings ist das Liften ein sehr aufwendiger Prozess und daher nicht praktikabel. Eine weitere Vorgehensweise ist die lokal gültigen Cuts für jeden Knoten zu speichern. Doch auch diese Möglichkeit, muss in der Praxis als zu aufwendig verworfen werden. Um dennoch einen Nutzen aus lokal gültigen Cuts zuziehen, wurden im Rahmen dieser Arbeit folgende Überlegungen angestellt.

Da die Cuts nicht sinnvoll gespeichert werden können, werden sie, direkt nachdem sie abgeleitet worden sind, wieder verworfen. Dabei stellt sich die Frage, welchen Nutzen diese Vorgehensweise haben könnte. Ist das betrachtete Teilproblem nach Anhängen der Cuts und erneutem Lösen vollständig untersucht, liegt der Vorteil auf der Hand. Wie kann jedoch die lokale Einschränkung durch Cuts an einem Knoten, der weiterverfolgt werden muss, sinnvoll genutzt werden?

Dazu soll folgendes Teilproblem innerhalb des Baumes betrachtet werden:

$$\begin{aligned} Z_{TP} = \min \quad & \sum_{j \in I} c_j x_j \\ & \bar{A}x \leq \bar{d} \\ & x_j \in \mathbb{Z}_+, \quad \forall j \in I \end{aligned}$$

wobei  $\bar{A}x \leq \bar{d}$ , die Originalrestriktionen  $Ax \leq d$  erweitert um die Verzweigungsrestriktionen ist.

Von diesem Modell werden Cuts abgeleitet, die nur lokal gültig sind. Dadurch ergibt sich u. U. ein neuer Zielfunktionswert  $Z_{TP+Cl}$ . Dieser Zielfunktionswert ist mindestens so groß, wie der des ursprünglichen Teilproblems,  $Z_{TP} \leq Z_{TP+Cl} \leq Z_{IP}$ . Das Modell wird wieder in seinen ursprünglichen Zustand zurückversetzt, die Cuts werden wieder aus dem Modell gelöst oder inaktiviert. So wird das Modell in der Kandidatenliste  $K$ , in der die noch zu bearbeitenden Knoten warten, gespeichert. Als Zielfunktionswert wird allerdings der des Teilproblems mit Cuts ( $Z_{TP+Cl}$ ) gespeichert. Infolge dessen ergibt sich ggf. schneller eine größere, globale Untergrenze.

Abbildung 6.2 und Abbildung 6.3 sollen die Relevanz der Ober- bzw. Untergrenze während des Branch-and-Cut Verfahrens verdeutlichen.

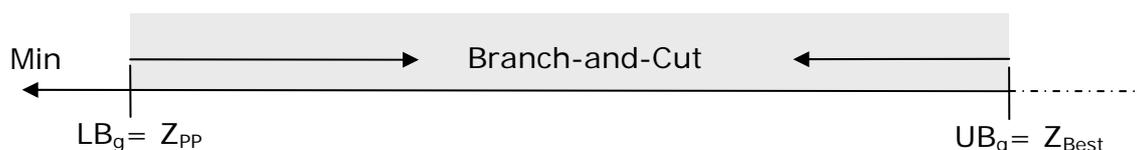


Abbildung 6.2: Ausgangspunkt des Branch-and-Cut Verfahrens

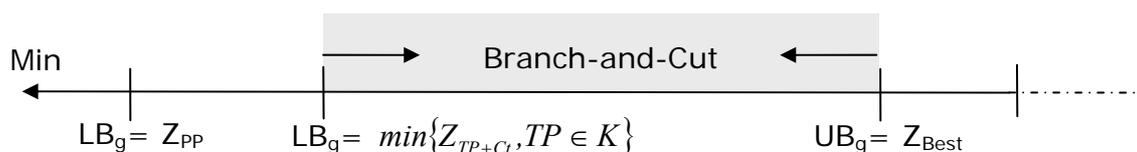


Abbildung 6.3: Darstellung der globalen Ober- bzw. Untergrenzen

Die globale Obergrenze ( $UB_g$ ) ist zu Beginn des Branch-and-Cut Verfahrens, wenn noch keine gültige, ganzzahlige Lösung bekannt ist,  $+\infty$ . Sobald eine ganzzahlige Lösung gefunden wird, wird dieser Wert aktualisiert. Somit ist der Wert der globalen Obergrenze immer gleichzeitig der Wert der bisher besten Lösung. Der Zielfunktionswert nach dem Supernode processing ( $Z_{PP}$ ) ist zu Beginn des Branch-and-Cut Prozesses die globale Untergrenze ( $LB_g$ ). Im Laufe des Verfahrens wird dieser Wert ersetzt, durch den kleinsten Zielfunktionswert der Teilprobleme, die noch in der Kandidatenliste warten. Je größer der Wert des Minimums innerhalb der Zielfunktionswerte in der Kandidatenliste ist, desto kleiner ist der anschließende Gap (vgl. Kapitel 4.4) zwischen Ober- und Untergrenze.

An einem Knoten wird allerdings bisher nicht überprüft, ob der Zielfunktionswert kleiner ist als die aktuelle Untergrenze. Dieser Fall konnte bisher nicht eintreten, da das Branching auf einer Variable nur eine Verschlechterung bewirken können.

Um sicherzustellen, dass die aktuelle Untergrenze auch tatsächlich nicht unterschritten wird, wird überprüft, ob der Zielfunktionswert jedes beliebigen Teilproblems mindestens so groß ist wie die aktuelle Untergrenze. Ist das nicht der Fall, wird das Modell um die Zielfunktion in Form folgender Restriktion

$$LB_g \leq \sum_{j \in I} c_j x_j$$

erweitert. Auf diese Art und Weise können lokal gültige Cuts sinnvoll innerhalb des Baumes verwendet werden. Diese beschriebene Vorgehensweise wurde mit lokal gültigen Gomory Cuts durchgeführt (vgl. dazu Kapitel 6.6.2.3).

## 6.6 Anwendung ausgewählter Cuts an einem Knoten

In den folgenden Abschnitten werden die bereits in Kapitel 4.7 vorgestellten Schnittebenen im Rahmen des Branch-and-Cuts Ansatzes betrachtet. Einige dieser Schnittebenen sind global gültig und können an jedem beliebigen Knoten abgeleitet werden. Die ggf. erforderlichen Anpassungen der Algorithmen werden beschrieben.

## 6.6.1 Cover Cuts

Ein Cover Cut wird von einer Originalrestriktion abgeleitet. In dieser Originalrestriktion können sowohl 0-1 Variablen sein als auch allgemeine Integer- oder kontinuierliche Variablen. Tatsächlich abgeleitet wird der Cover Cut allerdings von einer reinen 0-1 Restriktion. Wie in Kapitel 4.7.4.2 gezeigt, werden dazu die übrigen Variablen auf ihre Grenzen gesetzt. Im Rahmen des Branch-and-Cut Verfahrens können die Grenzen einer allgemeinen Integervariablen, wenn auf sie verzweigt wurde, verändert worden sein. Soll ein global gültiger Cover Cut abgeleitet werden, müssen die Originalgrenzen der allgemeinen Integervariablen verwendet werden. Wurden Binärvariablen innerhalb des Enumerationsbaumes auf null oder eins fixiert, müssen diese Variablen wie freie behandelt werden. Es besteht auch die Möglichkeit, die fixierten Variablen fixiert zu lassen und dann später in den Cover Cut zu liften. Diese Vorgehensweise entspricht allerdings genau dem, was ohnehin während der Generierung von Cover Cuts passiert. Die Variablen, die den LP-Lösungswert eins angenommen haben, werden erst nachdem ein Ausgangscover identifiziert wurde, in den Cut geliftet. Genauso werden die Variablen, die den LP-Lösungswert null angenommen haben, erst nachdem ein Ausgangscover identifiziert wurde, in den Cut geliftet.

Das Verfahren zur Identifizierung von Cover Cuts verwendet die aktuelle LP-Lösung, welche von den Verzweigungen beeinflusst ist. Sie wird jedoch nur zur Bestimmung der Reihenfolge, in der Variablen in einen Cover Cut aufgenommen werden, benutzt. Zusammenfassend ist festzuhalten, dass Cover Cuts global gültig sind und somit jederzeit an das Modell angehängen werden können.

## 6.6.2 Mixed Integer Gomory Cuts

### 6.6.2.1 Gemischt-ganzzahlige Modelle

Zur Ableitung des Mixed Integer Gomory Cuts wird das optimale Simplextableau als Grundlage herangezogen. Ein Mixed Integer Gomory Cut wird von der optimalen Basis abgeleitet, genauer gesagt von der Basisinversen. Wird ein Mixed Integer Gomory Cut an einem beliebigen Knoten des Baumes abgeleitet, wird die optimale Basis des entsprechenden Teilproblems

herangezogen. Die Basis eines Teilproblems ist aber u. U. keine zulässige Basis der Ausgangsaufgabe.

An jedem Knoten wird eine Variable  $x_j$ , für die Ganzzahligkeit gefordert wird, entweder nach oben oder nach unten beschränkt.

$$x_j \leq bu_j \quad , \quad x_j \geq bl_j$$

Zu jeder dieser Beschränkungen gehört eine logische Variable:

$$x_j + x_u = bu_j \quad \forall u \in F_u,$$

$F_u \subset I$  beschreibt die Menge der logischen Variablen, die zu einer oberen Beschränkung gehören und

$$x_j - x_l = bl_j \quad \forall l \in F_l,$$

$F_l \subset I$  beschreibt die Menge der logischen Variablen, die zu einer unteren Beschränkung gehören. Soll die Variable  $x_j$  auch Werte  $bu_j < x_j < ub_j$  annehmen können, wobei  $ub_j$  ihre Originalobergrenze ist, müssen die logischen Variablen der Verzweigungen negative Werte annehmen können. Wenn eine logische Variable einer Verzweigung Teil des Mixed Integer Gomory Cuts wird, ist dieser somit nur noch für alle die Teilprobleme gültig, für die auch die Verzweigung gültig ist.

*Beispiel:*

*Gegeben sei das Modell*

$$\max 7x_1 + 6x_2 - 22x_3$$

$$-8x_1 + 12x_2 - 30x_3 \leq 0$$

$$-4x_1 + 8x_2 - 50x_3 \leq 0$$

$$x_1 \leq 3$$

$$x_1, x_2, x_3 \in \mathbb{Z}_+$$

*Die Lösung der Relaxierung für dieses Modell ist  $x_1 = 3$   $x_2 = 2,33$   $x_3 = 0,13$ . Die Variable  $x_2$  soll als Branching-Variable ausgewählt werden und die Verzweigung  $x_2 \geq 3$  Teil des Modells sein.*

Dann ergibt sich folgendes Simplex- Tableau:

$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$	$x_8$	
								30,2
0	1	0	0	0	0	-1	0	3
0	0	1	-0,033	0	-0,267	-0,4	0	0,4
1	0	0	0	0	1	0	0	3
0	0	0	-1,667	1	-9,333	-12	0	8
0	0	0	0	0	0	1	1	0

Aus der 2-ten Zeile, in der die Integer Variable  $x_3$  in der Basis ist und mit 0,4 einen fraktionellen Wert hat, kann folgender Mixed Integer Cut abgeleitet werden.

$$0,022x_4 + 0,178x_6 + 0,4x_7 \geq 0,4$$

bzw.

$$0,002x_1 + 0,136x_2 + 0,66x_3 \geq 1,066$$

Die optimale Lösung für obiges Problem ist  $x_1 = 3$ ,  $x_2 = 1$ ,  $x_3 = 0$ . Durch den am Knoten generierten Cut würde diese Lösung abgeschnitten werden.

Der abgeleitete Mixed Integer Gomory Cut schneidet die optimale Lösung ab. Wie oben dargestellt, liegt das an den logischen Variablen der Verzweigungsrestriktionen.

Soll ein global gültiger Cut an das Modell angehängt werden, bleibt nur die Möglichkeit den lokal gültigen Cut, wie in Kapitel 6.4 beschrieben, zu liften. Dazu muss der Cut zunächst so umgeformt werden, so dass er nur aus Strukturvariablen besteht. Danach wird ein Koeffizient für die Variablen gesucht, die beschränkt wurden.

Leider ist das Liften von Mixed Integer Gomory Cuts in gemischt-ganzzahligen Modellen sehr aufwendig. Zur Ermittlung eines Lifting-Koeffizienten müssen, jeweils mehrer Probleme gelöst werden (vgl. Kapitel 6.4).

Werden Mixed Integer Gomory Cuts innerhalb des Baumes abgeleitet und nicht geliftet, sind sie lokal gültig und müssen wieder aus dem Modell entfernt werden.

Diese Vorgehensweise ist sehr aufwendig, da Gomory Cuts immer nur für eine LP-Optimierung abgeleitet werden. Demgegenüber steht der Vorteil, dass die i. d. R. sehr dicht besetzten Gomory Cuts an anderen Knoten nicht die LP-Lösungszeit beeinträchtigen.

Wird von dem Szenario ausgegangen, dass Gomory Cuts an einem Knoten abgeleitet werden und direkt wieder verworfen werden, sollte ein Augenmerk darauf gelegt werden, wann diese kurzfristige Einschränkung des Lösungsraumes überhaupt die Gesamtlösungszeit verbessern kann. Das ist prinzipiell nur dann der Fall, wenn ein Teilproblem, durch das Anhängen von Gomory Cuts vollständig untersucht ist. D.h., wenn es entweder unzulässig ist, ganzzahlig oder, wenn die Lösung schlechter ist als eine bereits gefundene globale Obergrenze. Muss nach Lösung des Teilproblems mit Gomory Cuts weiter verzweigt werden, war das Anhängen der Cuts zunächst scheinbar nutzlos. Sie sind zwar auch für die Folgeknoten des Teilproblems gültig, aber das würde das aufwendige Speichern der Cuts passend zu einem Knoten voraussetzen. Dennoch kann sich durch das Ableiten von lokal gültigen Gomory Cuts innerhalb des Baumes ein weiterer Vorteil ergeben auf den in Kapitel 6.6.2.3 eingegangen werden soll.

### 6.6.2.2 Reine 0-1-Modelle

Einen Sonderfall bilden Modelle, bei denen alle ganzzahligen Variablen Binärvariablen sind. Wie in [BCCN96] bewiesen werden konnte, können in diesem Fall theoretisch global gültige Mixed Integer Gomory Cuts abgeleitet werden.

Wird eine Binärvariable an einer Verzweigung ausgewählt, werden die zwei neuen Teilprobleme gebildet, indem die Binärvariable auf 0 bzw. 1 fixiert wird. Es wird hier davon ausgegangen, dass zur Fixierung einer 0-1-Variablen deren Obergrenze (Untergrenze) auf null (eins) gesetzt wird. Des Weiteren wird angenommen, dass die Menge der auf 1 fixierten Variablen leer ist  $F_u = \emptyset$ . Dazu werden alle Variablen  $x_j$ , die auf 1 fixiert wurden durch  $x'_j = 1 - x_j$  ersetzt. Alle fixierten Variablen sind nicht Teil der Basis.

*Beispiel:*

*Gegeben sei folgendes Modell:*

$$\max 7x_1 + 6x_2 - 22x_3$$

$$-8x_1 + 12x_2 - 30x_3 \leq 0$$

$$-4x_1 + 8x_2 - 50x_3 \leq 0$$

$$x_1, x_2, x_3 \in \{0,1\}$$

*Optimal gelöst ist dieses Modell, wenn  $x_1 = 1$ ,  $x_2 = x_3 = 0$*

*Betrachten wir aber zunächst die Lösung der Relaxierung*

$$x_1 = 1, x_2 = 0,778, x_3 = 0,044$$

*Variable  $x_2$  wird als Branching-Variable ausgewählt und auf eins fixiert.  $x_2$  kann aus dem Modell eliminiert werden, so dass sich dann folgendes Simplextableau ergibt:*

$x_1$	$x_3$	$x_4$	$x_5$	
-0,267	1	0,033	0	0,133
-9,3330	0	-1,667	1	2,667

*In der ersten Zeile ist die Binärvariable  $x_3$  in der Basis und hat einen fraktionellen Wert. Für diese Zeile kann der Mixed Integer Gomory Cut*

$$-0,0410x_1 - 0,0051214x_4 \geq 0,1333 \text{ abgeleitet werden.}$$

*Durch Ersetzen von  $x_4$  entsteht daraus der Cut*

$$0,1536x_3 \geq 0,1536 \text{ bzw. } x_3 \geq 1$$

*Da in der optimalen Lösung  $x_3 = 0$  ist, wird diese durch den Cut abgeschnitten.*

Dieser Mixed Integer Gomory Cut ist nicht global gültig. Das Problem ist, dass die fixierte Variable aus dem Modell entfernt wurde. Wird der Cut abgeleitet ohne die fixierte Variable, muss im Nachhinein dafür gesorgt werden, dass diese mit einem entsprechenden Koeffizienten in den Cut geliftet wird.

Alternativ dazu können die fixierten Variablen beim Ableiten des Gomory Cuts Teil des Modells bleiben. Dann bekommen sie einen Koeffizienten und der Gomory Cut ist global gültig.

Es wurde davon ausgegangen, dass keine fixierten Variablen in der Basis sind. Daraus folgt, dass die Basis am Verzweigungsknoten auch gleichzeitig eine zulässige Basis für das Originalproblem ist. Es handelt sich dabei nicht um eine optimale Basis für die Originalaufgabe, was aber auch nicht erforderlich ist.

*Betrachten wir noch einmal das obige Beispiel. Bleibt die Variable  $x_2$  Teil des Modells, wird also nicht durch die Fixierung auf eins entfernt sondern weiter wie jede andere Variable behandelt, sieht die erste Zeile in der  $x_3$  Basisvariable ist folgendermaßen aus*

$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	
-0,267	0,4	1	0,033	0	0,133

Aus dieser Zeile lässt sich der Cut

$$-0,0410x_1 - 0,09231x_2 - 0,0051214x_4 \geq 0,1333$$

ableiten.

Der Cut in Strukturvariablen ausgedrückt

$$-0,1538x_2 + 0,1538x_3 \geq 0 \text{ bzw. } -x_2 + x_3 \geq 0$$

verdeutlicht, dass die optimale Lösung nicht abgeschnitten wird, der Cut ist global gültig.

Zusammenfassend ist festzuhalten, dass ein Mixed Integer Gomory Cut für ein reines 0-1-Modell an jedem beliebigen Knoten abgeleitet werden kann. Es muss lediglich darauf geachtet werden, dass die auf null oder eins fixierten Variablen nicht in der Basis sind und nicht aus dem Modell entfernt werden.

### 6.6.2.3 Lokal Gültige Gomory Cuts

Da der Speicheraufwand zu groß ist, für jeden einzelnen Knoten die zu ihm gehörenden Gomory Cuts zu speichern, bleibt nur die Möglichkeit die lokal gültigen Cuts gleich wieder aus dem Modell zu entfernen. Diese Vorgehensweise kann unterschiedliche Vorteile hervorrufen. Ist der Knoten durch das Anhängen von Gomory Cuts unlösbar geworden, wurde eine ganzzahlige Lösung gefunden oder übersteigt der Wert der gefundenen Lösung den der globalen Obergrenze, ist der Nutzen offensichtlich: es muss nicht weiter verzweigt werden. Tritt keiner der drei Fälle ein, muss weiter verzweigt werden. Dazu werden die lokalen Gomory Cuts wieder aus dem Modell entfernt und das Problem mit dem neuen Zielfunktionswert gespeichert. Der Nutzen dieses neuen Zielfunktionswertes soll wie folgt realisiert werden.

Da die Gomory Cuts wieder entfernt wurden ist der Fall denkbar, dass ein Nachfolgerknoten einen kleineren Zielfunktionswert hat als sein Vorgänger. Da die entfernten Gomory Cuts aber für die Nachfolgerknoten theoretisch auch gültig sind nur nicht gespeichert wurden, kann der Zielfunktionswert des Vorgängerknotens als Untergrenze für den Zielfunktionswert des

Folgeknotens dienen. D.h. die Zielfunktion wird als Restriktion an das Modell angehängt und somit der Lösungsraum des Folgeknotens ggf. beschränkt.

Es stellt sich die Frage, wann lokale Gomory Cuts im Baum abgeleitet werden sollten. Da sie dazu dienen die Untergrenze zu erhöhen, bietet es sich an, Gomory Cuts nur an den Knoten abzuleiten, deren Zielfunktionswert in der Nähe der aktuellen Untergrenze liegt oder sogar nur an den Knoten, deren Zielfunktionswert die aktuelle Untergrenze darstellt.

### 6.6.3 Clique Cuts

Clique Cuts werden wie in Kapitel 4.7.6 beschrieben auf Basis von Cliques abgeleitet. Diese Cliques gehen aus dem Originalmodell hervor. Sie werden einmal vor dem Branch-and-Cut Prozess, ermittelt und in einer Tabelle abgespeichert. Bei der tatsächlichen Identifizierung von Clique Cuts werden lediglich diese Cliques daraufhin durchsucht, ob eine Verletzung gefunden werden kann. Dazu wird zwar die aktuelle Lösung der LP-Relaxierung benutzt, entscheidend ist jedoch, dass die Cliques global gültig sind. Jeder Clique Cut, der aus einer global gültigen Clique hervorgeht, ist global gültig.

Kritischer wird es, wenn Cliques innerhalb des Baumes ermittelt werden sollen. Das Verzweigen auf allgemeine Integervariablen stellt kein Problem dar. Bei der Identifizierung von Cliques werden sie auf ihre Grenzen gesetzt, dabei ist lediglich darauf zu achten, dass sie auf ihre Originalgrenze und nicht die Ober- oder Untergrenze der Verzweigungen gesetzt werden. Daraus folgt dann allerdings auch, dass keine neuen Cliques identifiziert werden können.

Binärvariablen werden auf ihre Ober- bzw. Untergrenze fixiert. Diese Fixierung kann natürlich wieder ignoriert werden. Dann ergeben sich aber wiederum keine anderen Cliques als am Ausgangsknoten. D.h. die Suche nach neuen Cliques innerhalb des Baumes ist nur sinnvoll, wenn 0-1-Variablen fixiert bleiben oder die durch das Verzweigen veränderten Grenzen der allgemeinen Integervariablen benutzt werden. Ein Cut, der aufgrund einer ggf. neu identifizierten Clique im Baum abgeleitet werden kann, ist nicht global gültig. Die Clique basiert auf bestimmten Modifikationen, die innerhalb des Baumes vorgenommen wurden. Diese Modifikationen werden ggf. wieder aufgehoben. Bleibt der Clique Cut dann Teil des Modells, schneidet er u. U. die optimale Lösung ab.

Ein lokal gültiger Clique Cut kann durch Liften global gültig gemacht werden. Im Folgenden soll zunächst der Fall eines reinen 0-1 Modells, von dem eine einfache Clique (Kapitel

4.7.6.1) an einem Knoten im Baum abgeleitet wird, betrachtet werden. Für alle Variablen, die in der Originalrestriktion von der diese Clique abgeleitet wird, fixiert wurden, muss ein Koeffizient gefunden werden. Bei der Ermittlung dieses Koeffizienten muss nur die Originalrestriktion in Betracht gezogen werden. Dabei muss ein Knapsack Problem (Kapitel 3.4.3) gelöst werden. Prinzipiell entspricht das Lifting-Verfahren dem, welches schon für die Verschärfung von Cover Cuts angewandt wurde (Kapitel 4.7.4.1.2). Der einzige Unterschied besteht darin, dass der Lifting-Koeffizient, sowohl wenn er positiv ist als auch wenn er negativ ist, benutzt wird. Das liegt daran, dass es hier in erster Line darum geht, den Cut global gültig zu machen und nicht einen schärferen Cut zu entwickeln.

Es wurde der folgende einfache Clique Cut im Baum abgeleitet:

$$\sum_{i \in C} \alpha_i x_i \leq \alpha_0$$

Diese Clique ist nur lokal gültig, da die Variable  $x_j$  auf eins fixiert ist. O. B. d. A. wird davon ausgegangen, dass alle Koeffizienten der Originalrestriktion positiv sind.

Um die Clique global gültig zu machen muss das Problem

$$\begin{aligned} g_j &= \max \sum_{i \in C} \alpha_i x_i \\ \sum_{i \in C} a_i x_i &\leq b \\ x_i &\in \{0,1\} \end{aligned}$$

gelöst werden. Wobei  $b$  die rechte Seite der Originalrestriktion, von der die Clique abgeleitet wurde ist, bevor  $x_j$  fixiert wurde. Die Menge  $C$  beinhaltet die Variablen, die in der Clique sind,  $a_i$  sind die Koeffizienten der Originalrestriktion und  $\alpha_i$  die Koeffizienten der Variablen im Clique Cut (diese sind i. d. R. eins außer eine andere Variable wurde bereits in den Clique Cut geliftet).

$\beta_j$  sei dann der gesuchte Lifting-Koeffizient, der sich durch  $\beta_j = g_j - \alpha_0$  ergibt.

Der geliftete global gültige Clique Cut ist dann:

$$\sum_{i \in C} \alpha_i x_i + \beta_j x_j \leq \alpha_0 + \beta_j$$

Wurde eine Variable auf null fixiert, kann keine global gültige Clique identifiziert werden, die nicht auch schon am Ausgangsknoten identifiziert werden konnte. Es gibt nur einen Fall, in dem es sinnvoll sein könnte eine auf null fixierte Variable zu liften. Variablen wurden fixiert und an einem Knoten konnte eine neue Clique identifiziert werden. In der Originalrestriktion, von der der Clique Cut abgeleitet wurde, ist eine auf null fixierte Variable. Diese Variable kann dann u. U. relativ unkompliziert in den Clique Cut geliftet werden. Da die Variable auf null fixiert ist, wird sie allerdings nicht zur Verletzung des Clique Cuts beitragen. Soll eine auf null fixierte Variable  $x_j$  dennoch geliftet werden, muss das Problem

$$g_j = \max \sum_{i \in C} \alpha_i x_i$$

$$\sum_{i \in C} a_i x_i \leq b - a_j$$

$$x_i \in \{0,1\}$$

gelöst werden und der gesuchte Lifting-Koeffizient ergibt sich aus  $\alpha_j = \alpha_0 - g_j$

An folgendem Beispiel soll das Liften von lokalen Clique Cuts verdeutlicht werden:

*Beispiel:*

*Gegeben sei die Ungleichung:*

$$24x_1 + 19x_2 + 14x_3 + 10x_4 + 5x_5 \leq 25$$

*Am Ausgangsknoten konnten von dieser Ungleichung die Cliques*

$$C_1 = \{1, 2, 3\}, C_2 = \{1, 2, 4\} \text{ und } C_3 = \{1, 5\} \text{ identifiziert werden.}$$

*Es soll nun der Knoten  $F_l = \{1\}$  und  $F_u = \{3\}$  betrachtet werden, an dem die optimale*

*LP-Lösung  $x^* = (0; 0; 1; 0,9; 0,4)^T$  ist.*

*An diesem Knoten kann von der obigen Ungleichung eine weitere Clique gefunden werden  $C_4 = \{2, 4, 5\}$*

*Bei dem dazugehörigen Clique Cut handelt es sich um einen lokal gültigen Cut, da er die optimale ganzzahlige Lösung  $x^{IP} = (0; 1; 0; 0; 1)^T$  abschneidet.*

*Durch Lösung des oben vorgestellten Knapsack-Problems*

$$\begin{aligned} \max \quad & x_2 + x_4 + x_5 \\ & 19x_2 + 10x_4 + 5x_5 \leq 25 \\ & x_2, x_4, x_5 \in \{0,1\} \end{aligned}$$

ergibt sich für  $x_3$  ein Lifting-Koeffizient von eins und der Clique Cut wird erweitert zu  $x_2 + x_3 + x_4 + x_5 \leq 2$ . Durch diesen Cut wird die relaxierte LP-Lösung abgeschnitten, die optimale IP-Lösung jedoch nicht mehr.

Wird auch für  $x_1$  das entsprechende Knapsackproblem gelöst, ergibt sich ein Lifting-Koeffizient von zwei. Ein global gültiger Clique Cut, der an diesem Knoten an das Modell angefügt werden kann, sieht also wie folgt aus:  $2x_1 + x_2 + x_3 + x_4 + x_5 \leq 2$

Handelt es sich nicht um ein reines 0-1 Modell oder um eine erweiterte Clique, von der ein Clique Cut abgeleitet wurde, muss das in Kapitel 6.4.2 beschriebene Lifting Verfahren angewandt werden. Dabei müssen alle allgemeinen Integervariablen der Originalrestriktion, auf die verzweigt wurde, geliftet werden. Beim Liften der allgemeinen Integervariablen in eine einfache Clique reicht die Originalrestriktion als Nebenbedingung aus. Kann für eine Variable kein Lifting-Koeffizient gefunden werden, muss der Cut verworfen werden.

Im Rahmen dieser Arbeit werden nur global gültige Clique Cuts im Baum abgeleitet.

### 6.6.4 Implikation Cuts

Bei den Implikation Cuts verhält es sich ähnlich wie bei den Clique Cuts. Auch Implikationen gehen aus dem originalen Modell hervor. Werden sie einmalig am ersten Knoten identifiziert sind sie global gültig. Das Ableiten von Implication Cuts besteht darin, die Implikationen daraufhin zu untersuchen, ob sie von der aktuellen LP-Lösung verletzt werden. Wenn Implication Cuts auf global gültigen Implikationen basieren, sind auch sie selber global gültig. Global gültig sind alle Implikationen, die vor dem Branch-and-Cut Verfahren identifiziert werden. Wenn das Originalmodell noch vorliegt, könnten theoretisch auch innerhalb des Baumes global gültige Implikationen aufgestellt werden. Das wären allerdings die gleichen die auch schon am ersten Knoten identifiziert werden konnten. Von dem veränderten Modell, in dem allgemeine Integervariablen beschränkt und 0-1 Variablen fixiert sind, können zwar ggf. neue Implikationen aufgestellt werden, diese sind aber nur noch lokal gültig. Dementsprechend

sind auch die Implication Cuts, die von solchen Implikationen abgeleitet werden nur lokal gültig. Sollen diese zur globalen Gültigkeit geliftet werden, muss das in Kapitel 6.4 vorgestellte Verfahren angewandt werden. Dabei müssen alle Variablen, auf die bis dahin verzweigt wurde, geliftet werden.

Im Rahmen dieser Arbeit werden Implication Cuts im Baum abgeleitet, die auf Implikationen beruhen, die am Ausgangsknoten aufgestellt wurden.

