4 Die strenge LP-Relaxierung

4.1 Lösung gemischt-ganzzahliger Modelle

An dieser Stelle soll kurz die Vorgehensweise beschrieben werden, die zur Anwendung kommt, wenn ein gemischt-ganzzahliges Modell gelöst werden soll. Dazu durchläuft das Modell in der Regel vier Phasen. Diese Phasen werden in Abbildung 4.1 als Überblick dargestellt.

Gegeben sei das ganzzahlige Optimierungsproblem

$$\begin{aligned} &\mathit{min} \quad \sum_{j \in I} c_j x_j \\ &Ax \le d \\ &x_j \in \mathbb{Z}_+ \qquad \forall j \in I \end{aligned} \qquad \mathit{(IP)}$$

Für die Definitionen vlg. Kapitel 3.2. und es sei $\mathbb{Z}_+ = \{x \in \mathbb{Z} \mid x \ge 0\}$ und $\mathbb{R}_+ = \{x \in \mathbb{R} \mid x \ge 0\}$.

Zu Beginn wird das Modell *relaxiert* gelöst. Die Relaxierung dieses Modells erfolgt, indem die Bedingung $x_j \in \mathbb{Z}_+$ aufgehoben und durch $x_j \in \mathbb{R}_+$ ersetzt wird. D.h. in der Lösung der Relaxierung von *(IP)* dürfen alle Variablen reelle Werte annehmen.

Die Lösung des relaxierten Modells dient als Grundlage für die nächste Phase, das sogenannten *Supernode processing*. Während des Supernode processings kommen verschiedene Techniken zum Einsatz, die zum Ziel haben, den Lösungsraum der relaxierten Lösung einzuschränken. Das erneuten lösen des Modells wird als *strenge* LP-Relaxierung (vgl. Kapitel 4.2) bezeichnet.

Dem folgend werden *Heuristiken* eingesetzt. Mit der Hilfe von Heuristiken wird, durch wiederholtes Runden und der Begrenzung von Variablen, die Komplexität des Modells reduziert, in der Hoffnung ganzzahlige Lösungen für ein Problem zu finden. Leider kann für eine gefundene, ganzzahlige Lösung in der Regel nicht bestimmt werden, ob es sich dabei um eine optimale Lösung handelt. Deshalb werden Heuristiken bei einem exakten Lösungsansatz nur als Hilfsmittel verwendet. Die ganzzahligen Lösungen, die durch Heuristiken gewonnen werden, dienen als Grenze für die weitere Suche nach der optimalen Lösung. Dank einer solchen Grenze können ggf. Variablen fixiert werden oder das anschließende Branch-and-Cut Verfahren kann verkürzt werden. Auch innerhalb des Branch-and-Cut Ansatzes können Heuristiken angewandt werden. Dabei soll eine in der Nähe des Knotens liegende ganzzahlige Lösung gefunden werden. Auf einzelne Heuristiken wird in dieser Arbeit nicht weiter eingegangen, es sei jedoch auf die Arbeiten [BCDM01; Chri05; Frie07; GlLa97] verwiesen.

Die abschließende Phase, das *Branch-and-Cut* Verfahren, ermittelt die optimale gemischt-ganzzahlige Lösung, immer vorausgesetzt es existiert eine. Im Laufe dieses Verfahrens müssen immer wieder Teilprobleme relaxiert gelöst werden. Auch ausgewählte Verfahren, des Supernode processings, werden an bestimmten Knoten wieder eingesetzt.

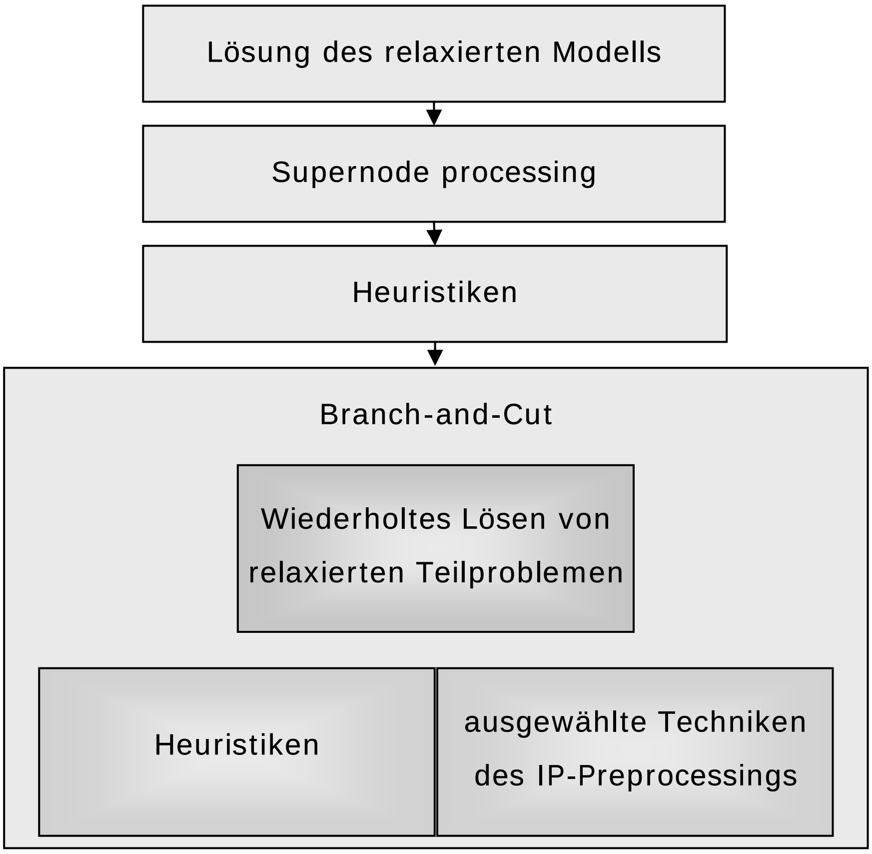

Abbildung 4.1: Das Verfahren zum Lösen von gemisch-ganzzahligen Modellen

4.2 Bedeutung der strengen LP-Relaxierung

Von einer *strengen* LP-Relaxierung ist die Rede, wenn durch Reformulierung des IP-Modells ein äquivalentes IP-Modell bestimmt wird, dessen LP-Relaxierung einen kleineren zulässigen Bereich als die LP-Relaxierung des Originalmodells aufweist. Das Modell kann sowohl manuell als auch automatisch reformuliert werden.

Es soll beispielsweise ausgedrückt werden, dass $y = 1$ sein muss, immer dann wenn $x = 1$ ist $x, y \in \{0,1\}$.

Dieser Zusammenhang kann durch $x \leq y$, aber auch durch $x \leq 2y$, ausgedrückt werden.

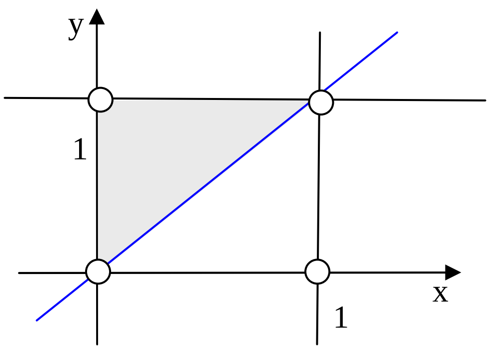

Abbildung 4.2: Formulierung $x \leq y$

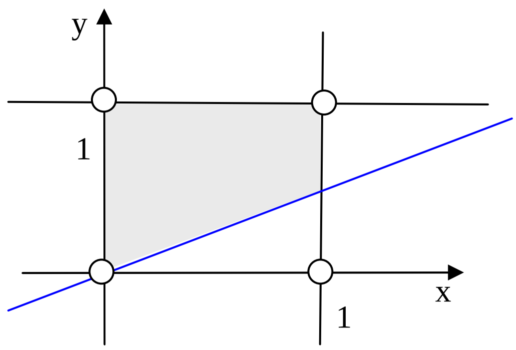

Abbildung 4.3: Formulierung $x \leq 2y$

Die beiden Abbildungen verdeutlichen, dass zwar beide Formulierungen logisch richtig sind, die Erste jedoch den Lösungsraum wesentlich mehr einschränkt als die Zweite.

Der Lösungsbereich des IP-Problems ist eine Teilmenge des Lösungsbereiches des relaxierten Problems. Jede ganzzahlige Lösung eines Problems ist immer auch eine gültige Lösung für das relaxierte Problem. Umgekehrt gilt das natürlich nicht. Trotzdem können einige Rückschlüsse auf eine ganzzahlige Ausgangsaufgabe gezogen werden, wenn diese relaxiert gelöst wird.

- Gibt es keine zulässige Lösung für die Relaxierung, dann kann es auch keine zulässige Lösung für das Originalproblem geben.
- Der Zielfunktionswert einer optimalen Lösung der Relaxierung liefert eine untere Grenze für die Originalaufgabe.

- Wenn die Lösung der Relaxierung ganzzahlig ist, obwohl die Bedingung der Ganzzahligkeit aufgehoben wurde, ist diese Lösung auch gleichzeitig die Lösung des Originalproblems.

Die Verkleinerung des Lösungsraums durch die strenge LP-Relaxierung führt i. d. R. dazu, dass weniger Zeit für die Suche nach einer optimalen IP-Lösung innerhalb des Branch-and-Bound/Cut Verfahrens benötigt wird. Durch das Konzept der strengen LP-Relaxierung [vgl. CrJP83; John89; Wols98] konnten große Fortschritte in der ganzzahligen Optimierung erzielt werden.

4.3 Manuelle Modellreformulierung

Durch die effiziente Formulierung eines Modells, kann der Lösungsraum in dem die IP-Lösung zu finden ist, eingeschränkt werden.

Die manuelle Reformulierung eines Modells ist sehr aufwendig. Für jedes Modell muss im Einzelnen dessen Struktur erfasst werden. Dieser Aufwand ist jedoch in einigen Fällen erforderlich, da z. T. durch die Modifikationen am Modell, dieses überhaupt erst gelöst werden kann oder zumindest die Lösungszeit erheblich reduziert werden kann. Im Regelfall wird versucht die Reformulierung des Modells mittels Algorithmen automatisch der eigentlichen Optimierung vorzuschalten (automatische Reformulierung (Kapitel 4.4)), das ist jedoch wie im Folgenden gezeigt werden soll leider nicht immer möglich.

Es soll von einer Formulierung der folgenden Art ausgegangen werden:

$$\sum_{j \in J} x_j \leq \sum_{j \in J} u_j y$$

Mehrere Variablen x_j, werden mit ihren entsprechenden Obergrenzen u_j durch eine Binärvariable y reguliert.

Obwohl diese Formulierung inhaltlich korrekt ist, besteht die Möglichkeit sie durch folgende Ungleichungen, in der der gleiche Sachverhalt disaggregiert ausgedrückt wird, besser darzustellen:

$$x_j \leq u_j y \qquad \forall j \in J$$

Es wird dabei von *schwachen* bzw. *scharfen* Formulierungen gesprochen.

Bei der schwachen, aggregierten Formulierung nimmt y in der relaxierten Lösung des Problems einen kleineren Wert an $y = \frac{\sum_{j \in J} x_j}{\sum_{j \in J} u_j}$. Wird hingegen die disaggregierte Formulierung gewählt nimmt y den größeren Wert $y = \min_{j \in J} \frac{x_j}{u_j}$ an. Welche Relevanz es hat, dass ein Modell disaggregiert formuliert wird, wird ausführlich in dem Artikel von [MWJS78] dargelegt.

Ein weiteres Beispiel, an dem deutlich wird, dass die Formulierung eines Modells sehr großen Einfluss auf dessen Lösbarkeit hat, ist im Rahmen der eindimensionalen Verschnittoptimierung zu finden. Die zugrunde liegende Anforderung ist dabei folgende: Es gibt Rollen einer bestimmten Breite. Aus diesen Rollen sollen von Kunden angeforderte kleinere Rollen geschnitten werden. Dabei soll möglichst wenig Verschnitt entstehen und somit der Bedarf aller Kunden mit einer Minimalzahl eingesetzter Rollen erfüllt werden. Inhaltlich korrekt kann diese Problemstellung mit folgendem Modell dargestellt werden:

$$
\begin{array}{lll}
\min & \sum_{j \in C} y_j & \\
& \sum_{i \in M} w_i x_{ij} \le r\, y_j & \forall j \in C \\
& \sum_{j \in C} x_{ij} = b_i & \forall i \in M \\
& y_j \in \{0,1\} & \forall j \in C \\
& x_{ij} \in \mathbb{Z}_+ & \forall j \in C, \forall i \in M
\end{array}
$$

Dabei ist M die Indexmenge der Kundenbedarfe, C die Indexmenge der gegebenen Rollen, r ist die Breite dieser Rollen, Ein Kunde $i \in M$ fordert b_i Rollen der Breite w_i. Die Entscheidungsvariable x_{ij} bestimmt die Anzahl der Rollen der Breite w_i, die aus der Rolle $j \in C$ geschnitten werden. Die Entscheidungsvariablen $y_j = 1$, wenn die Rolle j verwendet wird und 0 sonst. Diese Formulierung führt zu einer schlechten LP-Relaxierung. Viele der 0-1-Variablen werden fraktionelle Werte annehmen, da y_j auf den Wert der linken Seite $\sum_{i \in M} w_i x_{ij} \big/ r$ gesetzt wird.

Durch eine Umformulierung dieses Modells können entschieden bessere Ergebnisse erzielt werden. Die Idee dabei ist alle möglichen Schnittmuster zu generieren. Es sei S die Indexmenge der möglichen Schnittmuster. Durch a_{is} wird angegeben wie viele Rollen der Breite w_i aus dem Schnittmuster $s \in S$ hervorgehen. Jedem Schnittmuster wird eine ganzzahlige Entscheidungsvariable x_s zu zuordnen, die angeben soll, wie viele Rollen nach dem entsprechenden Schnittmuster geschnitten werden sollen. Es kann dann folgendes Modell aufgestellt werden:

$$min \quad \sum_{s \in S} x_s$$

$$\sum_{s \in S} a_{is} x_s = b_i \qquad \forall i \in M$$

$$x_s \in \mathbb{Z}_+ \qquad \forall s \in S$$

Die LP-Relaxierung dieser Formulierung kann sehr schnell gelöst werden. Bereits durch Rundung ergibt sich häufig eine sehr gute IP-Lösung. In der Arbeit [StKS04] wird ein Entscheidungsunterstützendes System zur Zuschnittsoptimierung von Rollenstahl vorgestellt, welches auf dieser Art und Weise der Formulierung der Problemstellung basiert.

Es kann darüber hinaus noch eine Verbesserung erzielt werden, wenn das Verfahren der Column Generation eingesetzt wird. Dieses Verfahren geht zurück auf [DaWo61; GiGo61; GiGo63]. Die Idee der Column Generation ist, nicht das gesamte Problem auf einmal zu lösen, sondern immer nur ein Teil des Gesamtproblems. Die bisher bei diesem Teilproblem fehlenden Variablen (Schnittmuster) werden, wenn sie zur Verbesserung der Lösung beitragen, nach und nach ergänzt. Bei der Column Generation Methode werden nach und nach nur die Variablen zu dem reduzierten Problem hinzugefügt, die möglicherweise den Zielfunktionswert verbessern. Diese Variablen werden anhand ihrer reduzierten Kosten identifiziert. Die reduzierten Kosten der Variablen die bisher nicht Teil des Modells sind, werden über die Dualvariablen des reduzierten Problems ermittelt. Hat eine Variable negative reduzierte Kosten, wird sie dem Problem hinzugefügt und das Problem wird erneut gelöst. Das Ganze wird solange fortgesetzt bis keine Variable mehr negative reduzierte Kosten hat.

Ein klassisches Beispiel, in dem die Column Generation sehr erfolgreich angewandt werden konnte, ist das Cutting Stock Problem [GiGo61; VBJN94; Vand96]. Darüber hinaus wurde die Column Generation Methode auch auf weitere bekannte Integerprobleme angepasst. Dazu zählen, u. a. das Generalized Assignment Problem [Save93], sowie Problemstellungen im

Zusammenhang mit der Tourenplanung, dem Crew Pairing und dem Crew Assignment [DeSo89; DeSD84; DeSS93]). In [BJNS98] wird neben einem Überblick über die Entwicklung der Column Generation Methode, eine allgemeingültige Vorgehensweise vorgestellt.

4.4 Automatische Reformulierung des Modells

Die strenge LP-Relaxierung basiert auf der Reformulierung des Originalmodells. Damit ein Modell verändert werden kann, muss die Struktur des einzelnen Modells nicht bekannt sein. Bei der automatischen Reformulierung kommen verschiedene Techniken zum Einsatz, die das gesamte Modell, nach bestimmten Kriterien untersuchen [RoWo87]. Ziel dieser Techniken ist es immer, den Lösungsraum der relaxierten Lösung einzuschränken. Dazu werden Modellkoeffizienten geändert, Variablen fixiert und Schnittebenen hinzugefügt. Diese automatische Reformulierung des Modells ist eine Art Vorbereitung, die die Suche nach der optimalen Integer Lösung erleichtern soll. Es wird daher auch vom *IP-Preprocessing* bzw. *Supernode processing* gesprochen.

4.5 Das Supernode processing

Unter dem Begriff Supernode processing werden verschiedene Verfahren zusammengefasst. Diese Verfahren werden auf das Originalmodell angewandt in der Hoffung, den möglichen Lösungsraum einschränken zu können. Sie basieren auf der Forderung nach Ganzzahligkeit für einige Variablen und sind implizit im Modell gegeben.

Als Maß für die Güte des Supernode processing dient die Veränderung des Zielfunktionswertes der LP-Relaxierung nach und vor dem Supernode processing. Ziel ist es den Zielfunktionswert durch das Supernode processing so zu verbessern, dass Zeit im Branch-and-Bound/Cut Prozess eingespart werden kann.

Durch die Reformulierung des Modells ist es möglich, das Branch-and-Bound/Cut Verfahren ggf. abzukürzen. Die Lösung des Modells nach dem Supernode processing schneidet einen Teil des Lösungsraumes ab.

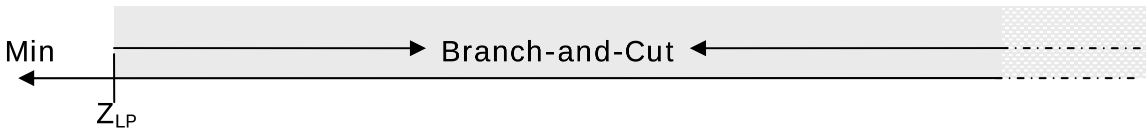

Abbildung 4.4: Lösungsraum ohne Supernode processing

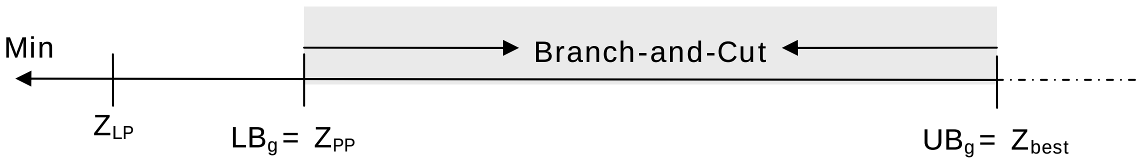

Abbildung 4.5: Lösungsraum mit Supernode processing

Dabei ist Z_{LP} der Zielfunktionswert des relaxierten Problems, Z_{PP} der Zielfunktionswert des relaxierten Problems nach dem Supernode processing und Z_{best} ist der Wert einer ganzzahligen Lösung. Der Wert des relaxierten Problems nach dem Supernode processing und der Wert einer ganzzahligen Lösung stellen die Untergrenze (LB_g) bzw. Obergrenze (UB_g) für den Bereich, in dem die optimale ganzzahlige Lösung liegen kann dar. Abbildung 4.4 und Abbildung 4.5 verdeutlichen, dass mit Hilfe des Supernode processings, dieser Bereich reduziert werden kann.

Um eine Vorstellung zu gewinnen, wie weit die Suche noch von der Lösung entfernt sein kann, wird ein sogenannter *Gap* berechnet. Es wird dabei zwischen relativem und einem absoluten Gap unterschieden. Mit absolutem Gap wird der Abstand zwischen der bisher besten gefundenen IP-Lösung und der aktuellen Untergrenze für das Problem bezeichnet. Beim relativen Gap wird diese Differenz noch ins Verhältnis gesetzt, indem sie durch die Obergrenze also die bisher beste IP-Lösung geteilt wird. Am Anfang stellt die LP-Lösung nach dem Supernode processing die Untergrenze dar, d.h. je höher diese ist, desto kleiner ist auch der Gap, der durch das Branch-and-Cut Verfahren geschlossen werden muss.

Die Einsparungen, die durch das Supernode processing im Branch-and-Bound/Cut Verfahren erzielt werden können, müssen immer ins Verhältnis gesetzt werden zu der Zeit, die dafür benötigt wird. Das Supernode processing und die dazu gehörenden Verfahren im Einzelnen sollten dementsprechend in irgendeiner Art und Weise eingeschränkt werden. Auf die möglichen Einschränkungen der einzelnen Techniken wird im Zuge ihrer Beschreibung eingegangen. Hier soll die Möglichkeit der Beschränkung des Supernode processings insgesamt aufgezeigt werden.

Tritt einer der folgenden Fälle ein, wird das Supernode processing abgebrochen.

- Die angewandten Techniken sind erschöpft. Es kann keine Veränderung mehr an dem Modell vorgenommen werden.
- Die Veränderung des Zielfunktionswertes unterschreitet einen bestimmten Wert. Damit soll die Effizienz der Techniken gemessen werden. Wenn der Wert der Zielfunktion sich nicht mehr oder kaum noch verbessert, d.h. an die IP-Lösung annähert, bewirken die Techniken zu wenig und das Supernode processing sollte abgebrochen werden.
- Das Supernode processing wurde k mal ausgeführt.

Zu den Techniken des Supernode processings gehören u. a. Verfahren zur Ableitung von Schnittebenen. Die Verfahren zur Ableitung von Mixed Integer Gomory, Cover, Implication und Clique Cuts können neben ihrem Einsatz im Supernode processing auch während des Branch-and-Cut Ansatzes verwendet werden. In Abschnitt 4.7 erfolgt eine detaillierte Beschreibung dieser Verfahren. Die Anwendung von Mixed Integer Rounding Cuts [MaWo01; Wess06] und Flow Cuts [GuNS99] erfolgt demgegenüber nur im Rahmen des Supernode processing.

Weitere Techniken des Supernode processing sind logische Tests, Bound Reduktionen, die Inaktivierung von Restriktionen und die einfache und erweiterte Koeffizienten Reduktion. Diese Techniken sind ausführlich in folgender Literatur beschrieben [JoKS85; SuSu99; SuSz94]. Zwei weitere Verfahren, die im Supernode processing zur Anwendung kommen sind die Fixed Charge Bound Reduction und Bound Reduktion ganzzahliger Variablen. Diese Verfahren, welche genauer in den folgenden Abschnitten beschrieben werden sollen, wurden, in der Hoffnung eine Verbesserung zu erzielen im Rahmen dieser Arbeit, implementiert und getestet. Dabei musste festgestellt werden, dass sie bei den meisten Modellen von anderen Verfahren dominiert werden.

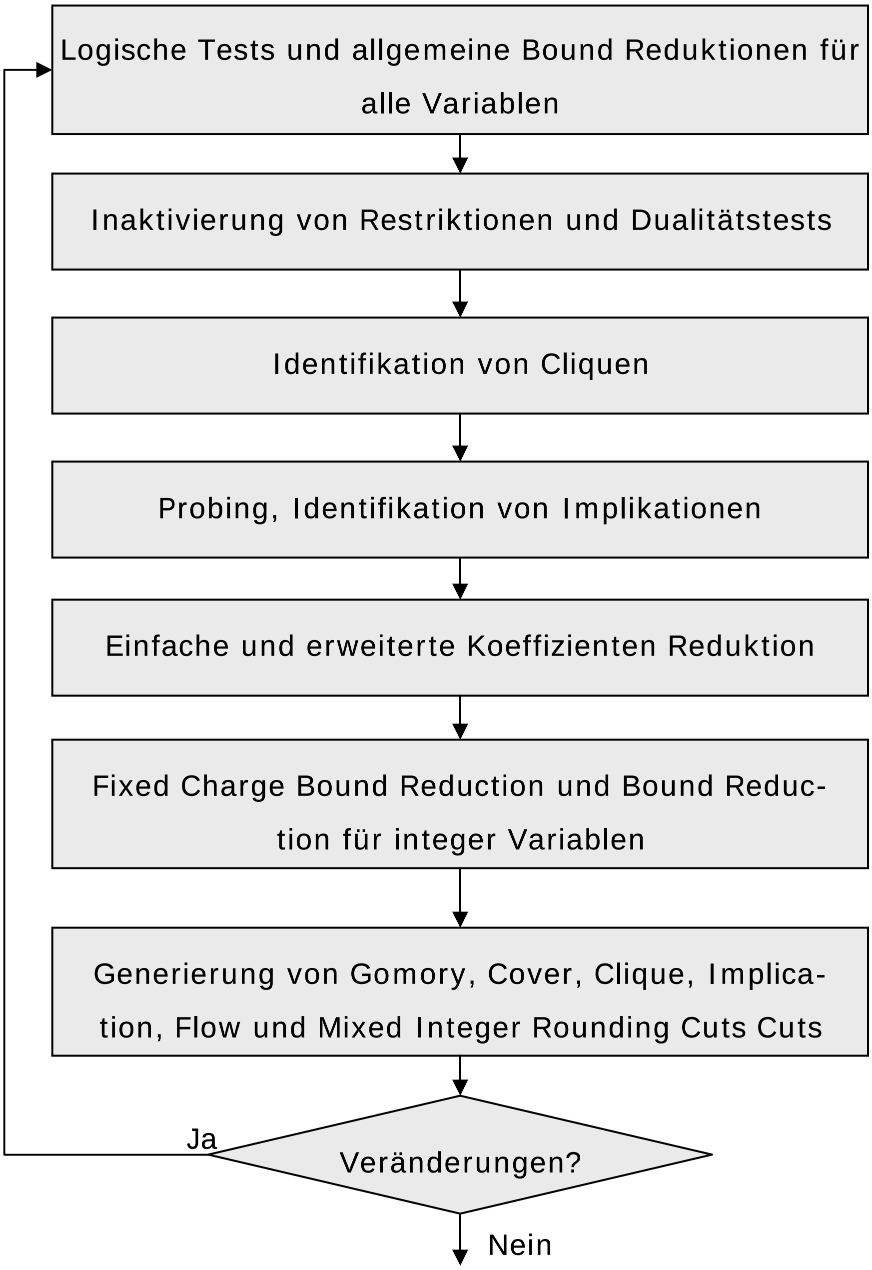

Abbildung 4.6: Ablauf des Supernode processings

Die Reihenfolge in der die verschiedenen Techniken ausgeführt werden, kann entscheidend sein für den Erfolg einer Technik. Es kann zu Substitutionseffekten kommen. Abbildung 4.6 gibt einen Überblick über die in der Optimierungssoftware MOPS (Kapitel 7.1) angewandten Techniken. Die dabei dargestellte Reihenfolge ist nicht zwingend und kann durch Parametereinstellungen geändert werden.

4.6 Bound Reductions

Es kann in zwei Arten von Bound Reductions unterschieden werden. Die eine Form der Bound Reduction basiert auf der Berechung der unteren und oberen Schranke einer Restriktion (vgl. [BrMW75; Save94]). Die andere Form der Bound Reduction, welche im Folgenden genauer beschrieben werden soll, basiert auf der Maximierung einer Restriktion oder einer einzelnen Variablen.

4.6.1 Fixed Charge Bound Reduction

In Kapitel 3.4.1 wurde die Klasse der Fixkostenprobleme (fixed charge problems) vorgestellt. Im Folgenden soll eine Vorgehensweise zur Verschärfung der LP-Relaxierung beschrieben werden, die auf diesen Fixkostenproblemen basiert [PaRW85; Suhl85].

Mathematisch können Fixkostenprobleme allgemein folgendermaßen beschrieben werden:

$$
\begin{aligned}
min \quad & \sum_{j \in J} c_j x_j \\
& Ax \leq d \\
& \left\{ \sum_{j \in J_g} a_{gj} x_j - M_{gf} x_f \leq 0 \right\} \subseteq Ax \leq d \; und \; g \in G, f \in B \\
& x_j \in \mathbb{R}_+ \qquad \forall j \in J \\
& x_f \in \{0,1\} \qquad \forall f \in B \subseteq J
\end{aligned}
\tag{4.1}
$$

Die Konstanten sind wie in Kapitel 3.2 definiert. B ist die Indexmenge der Fixkosten-Variablen und G die Indexmenge der Fixkosten-Restriktionen. $J_g \subset J,\ g \in G$. O.B.d.A wird davon ausgegangen, dass für $f \in B, j \in J_g, a_{gj} > 0,\ c_f \geq 0$ und $M_{gf} > 0$ gilt.

Für ein solches Problem ist es von besonderer Bedeutung, dass die *Fixed Charge Bounds*, hier *M*, so klein wie möglich sind. Wenn also *M* sehr groß ist, werden die Fixkosten bei der relaxierten Lösung des Problems ggf. ignoriert. Die Fixkosten-Variable nimmt einen sehr kleinen fraktionellen Wert an.

Die Fixed Charge Bound Reduction soll dazu führen, dass die Fixed Charge Bounds möglicherweise reduziert werden können. Dazu werden nacheinander die linken Seiten der Fixkosten-Restriktion maximiert. Nebenbedingung dieser Maximierung ist das gesamte ursprüngliche Modell und die zusätzliche Bedingung, dass die Zielfunktion einen mindestens genauso guten Wert annimmt wie eine bereits gefundene globale Obergrenze. Durch diese Maximierung wird der größtmögliche Wert ermittelt, den die Fixed Charge Bound annehmen kann, ohne dass sich eine schlechtere Gesamtlösung ergibt. Sei x_s mit $s \in B$ die Fixkosten-Variable für die Fixkostenrestriktion $r \in G$.

Folgendes Problem muss gelöst werden:

$$
\begin{aligned}
z'' = max \quad & \sum_{j \in J_r} a_{rj} x_j & \\
& Ax \leq d & \\
& \sum_{j \in J} c_j x_j \leq UB & \qquad (4.2) \\
& x_j \in \mathbb{R}_+ & \forall j \in J \\
& 0 \leq x_f \leq 1, x_f \in \mathbb{R}_+ & \forall f \in B \setminus \{s\}, x_s = 1
\end{aligned}
$$

wobei UB die globale Obergrenze ist, z.B. eine bereits gefundene, gültige, ganzzahlige Lösung für das Problem.

Die Fixkosten-Variable kann auf eins fixiert werden. Die Fixkosten-Restriktion ist Teil des Modells. Das Ergebnis der Maximierung würde zwangsläufig null sein, wenn $x_s = 0$. Da x_s eine Binärvariable ist, ist die Lösung der Maximierung folglich nur dann von Bedeutung, wenn die Fixkosten-Variable den Wert eins annimmt.

Gibt es keine Lösung für das Problem (4.2), kann x_s auf null fixiert werden. Denn dann gibt es auch keine Lösung für das Originalproblem, in der $x_s = 1$ ist. Die Fixierung von x_s auf eins ist die einzige, zusätzliche Bedingung, die das Problem unlösbar gemacht haben kann.

Angenommen das Problem ist lösbar und $z'' < M_{rs}$, dann kann M_{rs} durch z'' ersetzt werden.

Jede ganzzahlige Lösung des Originalproblems, die besser oder genauso gut wie UB ist, ist auch gültig für das modifizierte Problem, in dem M_{rs} durch z'' ersetzt wurde. Für den Beweis dieser Aussage sei auf [Suhl85] verwiesen.

4.6.2 Bound Reduction ganzzahliger Variablen

Die Idee durch eine Optimierung Grenzen zu reduzieren, kann auch auf jede einzelne ganzzahlige Variable angewandt werden.

Es wird folgendes gemischt-ganzzahliges Problem betrachtet.

$$
\begin{aligned}
\min \quad & \sum_{j \in J} c_j x_j \\
& Ax \le d \\
& lb \le x \le ub \\
& x_j \in \mathbb{R}_+ \qquad \forall j \in J \\
& x_j \in \mathbb{Z}_+ \qquad \forall j \in I \subseteq J
\end{aligned}
$$

Für die Definitionen vgl. Kapitel 3.2.

Durch die Maximierung bzw. Minimierung jeder ganzzahligen Variablen $s \in I$ kann ggf. deren Unter- bzw. Obergrenze angepasst werden. Das Originalproblem soll umgewandelt und relaxiert gelöst werden.

$$
\begin{aligned}
zz'' = \max \quad & x_s \\
& Ax \le d \\
& \sum_{j \in J} c_j x_j \le UB \\
& lb \le x \le ub \qquad\qquad (4.3) \\
& x_j \in \mathbb{R}_+ \qquad \forall j \in J
\end{aligned}
$$

wobei UB die globale Obergrenze ist, z.B. eine bereits gefundene, gültige, ganzzahlige Lösung für das Problem.

Diese Optimierung kann zu folgenden Ergebnissen führen:

- $ub_s > zz''$

  D.h., dass es keine gültige Lösung der Originalaufgabe gibt, die besser oder so gut wie eine bereits bekannte ganzzahlige Lösung (UB) ist, in der $x_s > zz''$ sein kann. Folglich kann die Obergrenze ub_s auf $\lfloor zz'' \rfloor$ herabgesetzt werden.

- $ub_s = zz''$

  Die Obergrenze der Variablen x_s kann nicht reduziert werden.

Soll die Untergrenze lb_s heraufgesetzt werden, muss aus dem Problem (4.3) ein Minimierungsproblem werden.

Dabei können die folgenden Ergebnisse auftreten:

- $lb_s < zz''$

  D.h., dass es keine gültige Lösung der Originalaufgabe gibt, die besser oder so gut wie UB_g ist, in der $x_s < zz''$ sein kann. Folglich kann die Untergrenze lb_s auf $\lceil zz'' \rceil$ heraufgesetzt werden.

- $ub_s = zz''$

  Die Untergrenze der Variablen x_s kann nicht angehoben werden.

Ein wichtiger Spezialfall ergibt sich mit Binärvariablen. Da x_s ausschließlich den Wert 0 oder 1 annehmen darf, kann x_s auf 0 fixiert werden, immer dann wenn das Ergebnis der Maximierung nicht eins ist. Genauso umgekehrt, nimmt x_s bei der entsprechenden Minimierung einen Wert größer null an, kann x_s auf 1 fixiert werden.

4.6.3 Die globale Obergrenze

Sowohl die Fixed Charge Bound Reduction, als auch die Bound Reduction ganzzahliger Variablen, basieren auf der Tatsache, dass eine globale Obergrenze vorliegen muss. Nur dann kann die Zielfunktion als Nebenbedingung an das Modell angehangen werden.

Jede gültige, ganzzahlige Lösung dient als eine globale Obergrenze des Problems. Im Folgenden sollen die Möglichkeiten, eine globale Obergrenze zu ermitteln, zusammengefasst werden.

- Die Anwendung von Heuristiken kann zu einer gültigen, ganzzahligen Lösung führen. Wird durch eine Heuristik eine IP-Lösung gefunden, können die Verfahren problemlos im Rahmen des Supernode processings angewandt werden.

- Dem Anwender ist eine globale Obergrenze bekannt. Diese kann entweder eine ihm aus Erfahrungswerten bekannte ganzzahlige Lösung sein oder ein Wert, von dem er weiß, dass er von der optimalen IP-Lösung nicht überschritten wird.
- Ansonsten besteht nur noch die Möglichkeit, eine globale Obergrenze zu schätzen. Auf Basis des LP-Zielfunktionswertes wird ein Wert für die globale Obergrenze bestimmt. Dabei besteht allerdings die Gefahr, die optimale Lösung abzuschneiden.
- Denkbar ist noch die globale Obergrenze aus dem Branch-and-Bound/Cut Prozess zu bekommen. Wird eine IP-Lösung im Baum gefunden, muss allerdings zum Ausgangsknoten zurückgekehrt werden, um die beiden Techniken anwenden zu können.

In MOPS werden beide Verfahren, wenn überhaupt, nur angewandt, wenn eine gültige, ganzzahlige Lösung während einer der Heuristiken gefunden wurde oder dem Anwender eine gültige ganzzahlige Lösung bekannt ist.

4.6.4 Steuerung der Bound Reductions

Insbesondere die Bound Reduktion ganzzahliger Variablen kann mit einem sehr hohen Zeitaufwand verbunden sein. Für jede ganzzahlige Variable müssen ein oder sogar zwei lineare Probleme gelöst werden. Damit das Verfahren nicht zu viel Zeit in Anspruch nimmt, wird eine Auswahl an ganzzahligen Variablen getroffen. Variablen, die in der relaxierten LP-Lösung nicht fraktionell sind, können zwar, müssen aber nicht im weiteren Verlauf fraktionell werden. Es bietet sich daher an, das Verfahren nur auf Variablen anzuwenden, die in der aktuellen relaxierten LP-Lösung fraktionell sind. Als beste Strategie hat es sich erwiesen, alle Variablen zu testen, die nicht quasi integer sind. Quasi integer ist eine Variable, die nicht integer ist aber nicht weiter als v von einem ganzzahligen Wert entfernt ist (v ist dabei ein sehr kleiner positiver Wert, beispielsweise 0,001).

Bei der Fixed Charge Bound Reduction verhält es sich ähnlich. Es hat sich auch bei diesem Verfahren als sinnvoll herausgestellt, es nur anzuwenden, wenn die Fixkosten-Variable in der relaxierten LP-Lösung nicht ganzzahlig ist, bzw. wenn sie nicht quasi integer ist. Davon abgesehen ist das Verfahren zusätzlich schon durch die Anzahl der Fixkosten-Restriktionen, die in einem Modell vorhanden sind, beschränkt.

Eine weitere Art der Steuerung der beiden Verfahren wird erreicht, in dem eine Obergrenze eingefügt wird, die angibt wie viele Iterationen maximal verbraucht werden dürfen. Ist diese Obergrenze erreicht werden die Verfahren abgebrochen. Handelt es sich um ein Modell, das relaxiert ohne viel Aufwand mit nur wenigen Iterationen gelöst werden kann, können auch die beiden Verfahren in größerem Umfang angewandt werden.

4.7 Schnittebenen (Cuts)

4.7.1 Allgemeine Erläuterungen zu Schnittebenen

Die Lösung eines ganzzahligen Optimierungsmodells kann erheblich erleichtert werden, indem Schnittebenen (*Cuts)* an das Modell angehangen werden [JüRT95; PoWo91].

Gegeben sei ein ganzzahliges Optimierungsproblem:

$$
\begin{aligned}
z = min \quad & \sum_{j \in I} c_j x_j \\
& Ax \leq d \\
& x_j \in \mathbb{Z}_+ \qquad \forall j \in I \qquad \text{(IP)}
\end{aligned}
$$

Mit P_{IP} wird die Menge der zulässigen Lösungen von (IP) und P_{LP} die Menge der Lösungen der Relaxierung von (IP) bezeichnet. Eine *gültige Ungleichung* (valid inequality) für (IP) z.B. $a^T x \leq a_0$, ist eine Ungleichung, die von allen zulässigen Lösungen von (IP) erfüllt wird, d.h. $a^T x \leq a_0$ gilt, wenn $x \in P_{IP}$.

Eine Schnittebene (Cut) ist dann eine gültige Ungleichung $a^T x \leq a_0$ für (IP), für die $a^T x > a_0$ für zumindest einige $x \in P_{LP}$ gilt. Durch das Anfügen von Schnittebenen an das Modell, wird die LP-Relaxation *verschärft*, d.h. der Lösungsraum der LP-Relaxierung wird eingeschränkt, ohne dass sich die Menge der zulässigen Lösungen von (IP) verändert.

Schnittebenen resultieren aus der Lösung des *Allgemeinen Separationsproblems*, welches sich folgendermaßen beschreiben lässt:

Der Vektor x^* stellt eine optimale Lösung der Relaxierung von (IP) dar mit $x^* \notin \mathbb{Z}_+$. Wird eine Klasse von gültigen Ungleichungen betrachtet und der Vektor x^*, besteht das allgemeine Separationsproblem darin, entweder eine gültige Ungleichung zu finden, die von x^* verletzt wird (d.h. eine Schnittebene ist), oder zu beweisen, dass in der betrachteten Klasse von Ungleichungen keine verletzte existiert.

Das Separationsproblem zum Auffinden von Schnittebenen kann exakt oder mittels einer Heuristik gelöst werden. Leider ist ein exakter Algorithmus nicht für alle Klassen von Schnittebenen bekannt oder es kann gezeigt werden, dass das Separationsproblem für bestimmte Klassen von Schnittebenen NP-schwer ist. Die Anwendung einer Heuristik hingegen garantiert nicht, dass wenn in einer Klasse von gültigen Ungleichungen keine verletzte Schnittebene aufgefunden werden kann, auch keine existiert.

Würde während der Lösung der LP-Relaxierung nicht die Ganzzahligkeitsbedingung der Variablen aufgehoben werden, wären Schnittebenen redundant.

Zur Veranschaulichung des Prinzips von Schnittebenen soll im Folgenden mit Abbildung 4.7 und Abbildung 4.8 eine grafische Interpretation eines ganzzahligen Optimierungsmodells herangezogen werden.

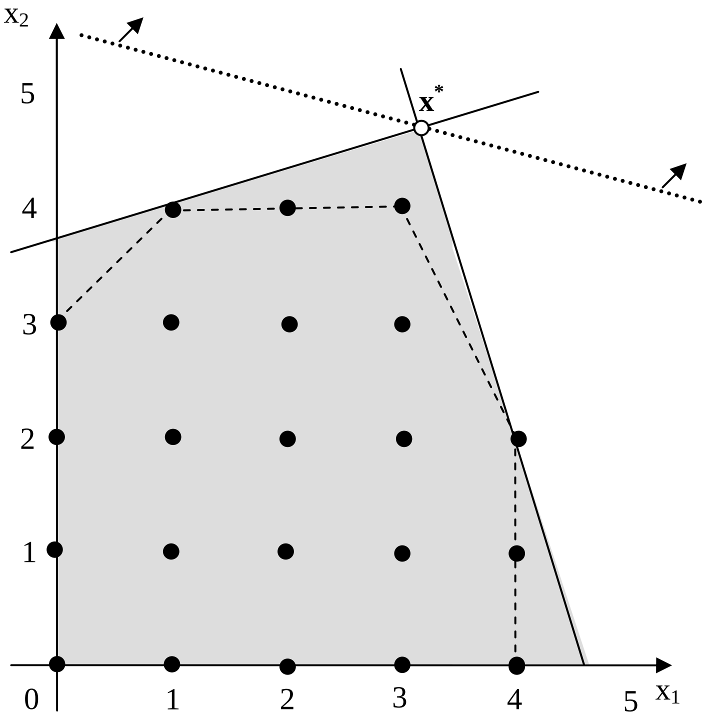

Abbildung 4.7: Graphische Darstellung eines ganzzahligen Optimierungsproblems

Der farbige Bereich entspricht dem Lösungsraum des relaxierten Problems. Die Punkte innerhalb dieses Bereiches stellen mögliche ganzzahlige Lösungen dar und die gestrichelten Geraden definiert die konvexe Hülle der ganzzahligen Lösungen. Die gepunktete Gerade soll die Neigung der Zielfunktion wieder geben. Der weiße Eckpunkt x^* stellt die Lösung des relaxierten Problems dar. Ziel ist es eine Schnittebene zu identifizieren, die den Lösungsraum des relaxierten Problems so einschränkt, dass die Lösung x^* nach erneutem Lösen der Relaxierung abgeschnitten wird. In Abbildung 4.8 wird so eine Schnittebene mit der dicken Geraden dargestellt. Da das ursprüngliche x^* nicht mehr Teil des zulässigen Lösungsraums ist, ergibt sich ein neues x^*.

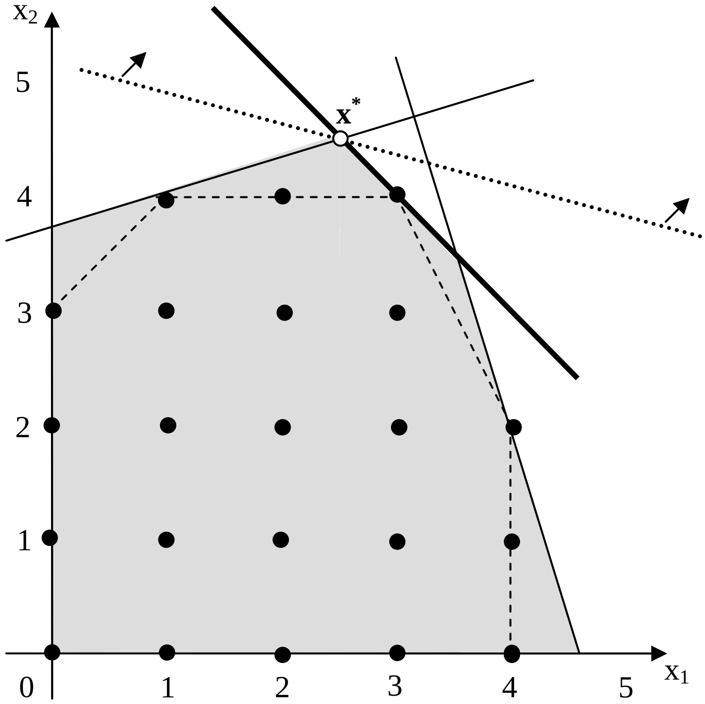

Abbildung 4.8: Graphische Darstellung eines ganzzahligen Optimierungsproblems mit Cut

Schnittebenen führen bei einem Maximierungsproblem dazu, dass der Zielfunktionswert der LP-Relaxierung sinkt.

Schnittebenen zeichnen sich durch folgende Eigenschaften aus:

- Sie werden von allen zulässigen ganzzahligen Lösungen erfüllt und sind ursprünglich nicht im IP-Modell vorhanden.
- Sie verkleinern i. allg. den Lösungsraum ohne dabei zulässige Lösungen des IP-Modells abzuschneiden.
- Sie schneiden eine optimale Lösung der LP-Relaxierung ab.

- Das Modell wird durch das Hinzufügen von Schnittebenen umfangreicher. Für die Lösung der LP-Relaxierungen ist es besser, Modelle mit möglichst wenig Restriktionen und Variablen zu haben. Durch das Hinzufügen der Schnittebenen wird i.d.R die Anzahl der zu lösenden Teilprobleme im Branch-and-Bound/Cut Verfahren reduziert. Dieser Vorteil überwiegt gegenüber dem Vorteil die LP-Relaxierungen schneller zu lösen.
- Es kann vorkommen, dass durch das Hinzufügen von Schnittebenen die optimale Lösung des IP-Modells gefunden wird.

Eine der zentralen Fragen bei der Generierung von Schnittebenen ist, ob von einer LP-Lösung eine einzelne Schnittebene oder mehrere abgeleitet werden, bevor reoptimiert wird.

Werden von einer LP-Lösung mehrere abgeleitet, kann es dazu kommen, dass u. U. Schnittebenen abgeleitet werden, die nach dem Anhängen einer einzelnen schon nicht mehr verletzt gewesen wären. Es muss also in Kauf genommen werden, dass das Modell durch Schnittebenen erweitert wird, welche den gleichen oder einen ähnlichen Bereich des Lösungsraums abschneiden. Das kann durch das ständige Reoptimieren nach dem Anfügen jeder einzelnen ausgeschlossen werden. Dafür ist die Anzahl der LP-Optimierungen, die durchgeführt werden müssen entschieden höher. Der Aufwand einer LP-Optimierung ist nicht zu unterschätzen, daher sollten mehrere ausgewählte Schnittebenen dem Modell hinzugefügt werden, bevor reoptimiert wird.

Dies leitet über zu einer weiteren entscheidenden Frage, nach welchen Kriterien diese ausgewählten Schnittebenen ausgesucht werden. Dabei gibt es für die verschiedenen Klassen von Schnittebenen spezifische Auswahlkriterien die bei der Vorstellung der einzelnen Schnittebenen diskutiert werden. Ein Kriterium das allen Schnittebenen gemein ist, ist die Qualität einer Schnittebene.

4.7.2 Qualität einer Schnittebene

Die Schnittebenen, die tatsächlich an das Modell angehangen werden, sollten von guter *Qualität* sein. Was ist mit dem Begriff Qualität im Umgang mit Schnittebenen zu verstehen?

Die konvexe Hülle aus den gültigen Lösungen eines gemischt-ganzzahligen Modells ist ein Polyeder. Die besten Schnittebenen sind die, die dazu benötigt werden, dieses Polyeder zu beschreiben. Im Folgenden sollen kurz einige Begriffe geklärt werden.

Die Lösungsmenge $P = \{x \in \mathbb{R}_+ : Ax \leq d\}$ eines linearen Ungleichungssystems ist ein konvexes Polyeder.

$\alpha x \leq \alpha_0$ ist eine gültige Ungleichung für *P*, wenn $\alpha x \leq \alpha_0, \quad \forall x \in P$ gilt.

S definiert eine Seite (Face) des Polyeders *P*, wenn $S = \{x \in P : \alpha x = \alpha_0\}$ gilt.

Mit *dim*(*P*) wird die Dimension eines Polyeders beschrieben. Die Dimension eines Polyeders ist gleich der maximalen Anzahl linear unabhängiger Vektoren im Polyeder.

Eine Facette *F* ist eine Seite des Polyeders, deren Dimension eins weniger ist als die Dimension des Polyeders, *dim*(*F*) = *dim*(*P*) – 1.Um ein Polyeder vollständig mit gültigen Ungleichungen zu beschreiben, muss jede Facette von einer gültigen Ungleichung dargestellt werden. Eine gültige Ungleichung $\alpha x \leq \alpha_0$ ist notwendig für die Beschreibung des Polyeders genau dann, wenn sie eine Facette definiert.

Am effektivsten sind also Schnittebenen, die eine Facette für das Polyeder des gemischt-ganzzahligen Modells definieren. Solche Schnittebenen können jedoch nur selten gefunden werden. Da auch Schnittebenen, die keine Facette definieren, die Lösungszeit eines gemischt-ganzzahligen Modells erheblich beschleunigen können und darüber hinaus sehr viel leichter zu identifizieren sind, sollen auch diese Schnittebenen an das Modell angehangen werden.

Eine Möglichkeit unter abgeleiteten Cuts auszuwählen wurde in [BaCC96; AnCF03] vorgestellt. Dabei wird die euklidische Entfernung zwischen der durch eine Schnittebene $\alpha x \leq \alpha_0$ hervorgerufenen Hyperebenen $\alpha x = \alpha_0$ und dem Punkt x^*, der die Lösung der LP-Relaxierung darstellt, verwendet:

$$ent(x^*, \alpha, \alpha_0) := \frac{\left|\alpha^T x^* - \alpha_0\right|}{\|\alpha\|} \tag{4.4}$$

$$\text{wobei } \|\alpha\| = \sqrt{\alpha_1^{\ 2}, \ldots, \alpha_n^{\ 2}} \text{ und } \alpha \neq 0$$

Je größer der Abstand ist, d.h. je weiter der Punkt x^* davon entfernt ist die Schnittebenen zu erfüllen, desto schärfer ist die Schnittebene.

Natürlich ist für die Qualität einer Schnittebene ausschlaggebend, dass sie von der aktuellen LP-Lösung verletzt wird. Wird die Formel (4.4) so modifiziert, dass im Zähler Max{ $\alpha^T x^* - \alpha_0, 0$} genommen wird, so weisen positive Werte dieses Ausdrucks daraufhin, dass es sich um eine verletzte Schnittebene handelt.

Als weiteres Maß für die Qualität einer Schnittebene kann die Veränderung des Zielfunktionswertes herangezogen werden. Allerdings kommt es gerade bei kleinen aber dafür sehr schwer zu lösenden Problemen vor, dass eine Schnittebene keine Verbesserung in der Zielfunktion hervorruft aber trotzdem entscheidet zur Lösung des Problems beiträgt. Des Weiteren kann auf diese Art und Weise die Qualität einer Schnittebene erst ermittelt werden, nachdem sie an das Modell angefügt wurde und nachdem das Modell reoptimiert wurde.

4.7.3 Liften von Schnittebenen

Im Folgenden soll das allgemeine L*iften* von Ungleichungen beschrieben werden. Beim Liften werden Variablen, die nicht Teil einer Ungleichung sind, mit in die Ungleichung aufgenommen. Dazu wird für den Koeffizienten einer solchen Variablen ein gültiger Wert ungleich Null gesucht. Ungleichungen sollen dadurch verschärft werden. Natürlich ist hierbei insbesondere darauf zu achten, dass die Ungleichungen gültig bleiben. Da es aus praktischen Gesichtspunkten zu aufwendig ist, alle Ungleichungen eines Modells zu liften, werden, wenn überhaupt, i. d. R. nur Schnittebenen geliftet.

Gegeben sei ein binäres Optimierungsproblem:

$$
\begin{aligned}
z = min \quad & \sum_{j \in B} c_j x_j \\
& Ax \le d \\
& x_j \in \{0,1\} \qquad \forall j \in B
\end{aligned}
\tag{IP}
$$

Mit $\sum_{j \in B} a_j x_j \le b$ wird eine gültige Ungleichung für (IP) beschrieben.

Mit PJ wird die Menge der zulässigen Lösungen von (IP) bezeichnet. Es sei $a_k = 0$ für ein $k \in B$. Beim Liften wird versucht, die Variable x_k in die Ungleichung einzubeziehen, d.h. einen Koeffizienten a_k ungleich null so zu bestimmen, dass die Ungleichung mit dem neuen

Koeffizienten a_k gültig bleibt und die neue Ungleichung die ursprüngliche dominiert. Eine mögliche Vorgehensweise besteht darin, für die Variable x_k die IP-Probleme (P1) und (P0) zu lösen:

$$z_k = max \sum_{j \in B-\{k\}} a_j x_j \text{ , } x \in PJ \wedge x_k = 1 \qquad \text{(P1)}$$

Falls es keine zulässige Lösung mit $x_k = 1$ gibt, dann kann x_k auf 0 fixiert werden. Ist $a_k^{'} = b - z_k > 0$, so ergibt sich unmittelbar, dass $a_k^{'} x_k + \sum_{j \in B-\{k\}} a_j x_j \leq b$ eine gültige Ungleichung definiert.

Das Optimierungsproblem (P0) ist wie folgt definiert:

$$z_k = max \sum_{j \in B-\{k\}} a_j x_j \text{ , } x \in PJ \wedge x_k = 0 \qquad \text{(P0)}$$

Insofern $x_k = 0$ zu keiner zulässigen Lösung führt, besteht die Möglichkeit, x_k auf 1 zu fixieren. Gibt es eine zulässige Lösung und ist dann $a_k^{'} = z_k - b < 0$, so ergibt sich unmittelbar, dass $a_k^{'} x_k + \sum_{j \in B-\{k\}} a_j x_j \leq b + a_k^{'}$ eine gültige Ungleichung definiert

Die Vorgehensweise bei (P1) wird als „Up-Liften auf eins“, die bei (P0) als „Up-Liften auf null“ bezeichnet. Jede beliebige 0-1-Variable deren Koeffizient in einer Ungleichung null ist, ist ein Lifting-Kandidat. Allerdings ist die Optimierung der IP-Probleme (P1) und (P0) aufwendig. Um trotzdem jede beliebige 0-1-Variable liften zu können, kann man die Probleme (P1) und (P0) als LP-Probleme lösen. Der Lifting-Koeffizient ergibt sich dann für LP(P1) aus $a_k^{'} = b - \lfloor z_k \rfloor$, wenn $a_k^{'} > 0$ und für LP(P0) aus $a_k^{'} = \lceil z_k \rceil - b$, wenn $a_k^{'} < 0$. Die so ermittelten Koeffizienten sind zwar ggf. kleiner, aber das Lösen der Probleme LP(P1) und LP(P0) ist weniger aufwendig.

Noch einfacher wird das Liften, wenn die zu liftende Ungleichung eine Schnittebene ist, die von genau einer Restriktion des Originalmodells abgeleitet wurde. Und die 0-1 Variable, die geliftet werden soll Teil dieser Restriktion des Originalmodells ist. In diesem Fall sind die IP-Probleme (P1) und (P0) binäre Knapsack-Probleme, weil nur die Originalrestriktion Nebenbedingung der Maximierung ist. Knapsack-Probleme lassen sich i. d. R. schnell lösen. Das Up-Lifting auf null (P0) ist dann allerdings nicht erforderlich, da o.B.d.A. alle Koeffizienten positiv sind und sich unter dieser Voraussetzung durch das Aufnullsetzen einer Variablen keine neuen Möglichkeiten ergeben. Diese Form des Liftens findet z.B. bei Cover Ungleichungen Anwendung. Die konkrete Vorgehensweise dazu wird ausführlich in Kapitel 4.7.4.1.2 beschrieben.

4.7.4 Cover Cuts

Cover Cuts wurden erstmalig Anfang der achtziger Jahre durch [CrJP83] erfolgreich eingesetzt. Mehrere 0-1-Modelle, die bis dahin als unlösbar galten, konnten daraufhin gelöst werden. Die theoretischen Grundlagen von Cover Ungleichungen wurden bereits in [Bala75; BaZe78; HaJP75; Wols75 und Zeme78] ausführlich untersucht und dokumentiert. Entscheidend ist allerdings eine effiziente und effektive Form der Ableitung von Cover Cuts. Der Veröffentlichung [CrJP83] folgend, wurden verschiede Möglichkeiten zur Herleitung von Cover Cuts vorgestellt (vgl. [HoPa91; GuNS98]). Die mathematischen Grundlagen der in dieser Arbeit vorgestellten Algorithmen basieren auf dem von [GuNS98] entwickelten Verfahren. Cover Cuts spielen vor allem für reine 0-1-Modelle eine wichtige Rolle.

Im Folgenden soll die Idee eines Covers Cuts dargestellt werden.

Gegeben sei folgendes 0-1-Modell:

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

Die Konstanten sind wie in Kapitel 3.2 definiert.

Sei $\sum_{j \in B} a_j x_j \leq b$ eine Restriktion aus $Ax \leq d$ von (IP). O.B.d.A. sei $a_j \geq 0$, da bei negativen Koeffizienten die Binärvariable durch ihr Komplement ersetzt werden kann $\tilde{x}_j = (1 - x_j)$.

Dann lässt sich ein Cover für diese Restriktion finden, wenn $\sum_{j \in B} a_j > b$. Das ist für jede nicht redundante Restriktion der Fall. Die Teilmenge *C* der Gesamtheit der Variablen der Restriktion stellt ein Cover dar, wenn gilt:

- $\sum_{j \in C} a_j > b$ Die Koeffizientensumme aller Variablen des Covers ist größer als die rechte Seite.

- $\sum_{j \in C} a_j - a_k \le b \quad \forall k \in C$ Wird ein beliebiger Koeffizient nicht mit aufsummiert, so ist die Summe der verbleibenden Koeffizienten kleiner oder gleich der rechten Seite.

Jedes Cover impliziert folgenden gültigen Cover Cut:

$$\sum_{j \in C} x_j \le |C| - 1.$$

Die Idee eines Cover Cuts soll zunächst an einem Beispiel verdeutlicht werden:

Beispiel:

Ein Cover für die Restriktion:

$$13x_1 + 15x_2 + 7x_3 + 11x_4 + 9x_5 + 5x_6 \le 26$$

ist $C = \{1,2,3,4,5\}$. *Aus diesem Cover lässt sich der folgende Cover Cut ableiten:*

$$x_1 + x_2 + x_3 + x_4 + x_5 \le 2.$$

Ein Cover Cut ergibt sich aus der Forderung, dass die Variablen nur den Wert 0 oder 1 annehmen dürfen. Es können maximal zwei Originalkoeffizienten der Variablen, die im Cover Cut vertreten sind, aufsummiert werden, ohne dass sie die rechte Seite überschreiten.

Die Variable x_6 *ist nicht Teil des Cover Cuts, da es möglich ist unter Einbeziehung von* x_6 *drei Variablen zu addieren, ohne die rechte Seite der Ungleichung zu übersteigen.*

Ist x^* eine Lösung des relaxierten Problems (IP) und $x^* \notin \{0,1\}^n$ dann soll, wenn möglich x^* mittels eines Cover Cuts abgeschnitten werden. Durch die Lösung des Separationsproblems für Cover Ungleichungen, soll ein Cover Cut gefunden werden, der durch die Lösung der LP-Relaxierung verletzt wird. Ein möglichst verletzter Cut kann durch die Minimierung von $|C| - 1 - \sum_{j \in C} x_j^*$ ermittelt werden. Eine Variable $z_i \in \{0,1\}$ soll $z_i = 1$ sein, wenn $i \in C$ und $z_i = 0$ sonst. Der zu minimierende Term lässt sich dann durch $\sum_{i \in B} (1 - x^*) z_i - 1$ ausdrücken.

Das zu lösende Separationsproblem für die Klasse von Cover Ungleichungen kann folgendermaßen beschrieben werden:

$$\gamma = min\left\{\sum_{i\in B}(1-x^{*})z_i \;:\; \sum_{i\in B} a_i z_i > b, z_i \in \{0,1\}\right\}$$

Ist $\gamma < 1$ wird x^* durch die Cover Ungleichung $\sum_{j\in C} x_j \leq |C| - 1$ abgeschnitten. Als Alternative zu dieser exakten Lösung des Separationsproblem für Cover Ungleichungen soll im Folgenden eine Heuristik beschrieben werden, die zu verletzten Cover Cut führt.

4.7.4.1 Vorgehensweise zur Identifizierung von Cover Cuts

Zum Auffinden eines Cover Cuts wird das in [GuNS98] beschriebene Verfahren zur Hilfe genommen. Durch einige Veränderungen an diesem Verfahren konnten noch bessere Ergebnis durch den Einsatz von Cover Cuts erzielt werden. Es kommen zwei verschiedene Sortierreihenfolgen für die Variablen, die einen fraktionellen Wert in der aktuellen LP-Lösung haben zur Anwendung. Bei der Identifizierung eines Covers spielt diese Sortierreihenfolge eine entscheidende Rolle. Eine weitere Verbesserung in dieser Arbeit geht damit einher, dass die während des Lifting-Prozess auftretenden Knapsack Probleme exakt gelöst werden.

Das Auffinden eines verletzten Cover Cuts erfolgt in zwei Schritten. Im ersten Schritt wird ein Cover ermittelt. Danach wird im zweiten Schritt versucht, das Cover zu liften.

4.7.4.1.1 Auffinden eines Covers

Grundlage der in [GuNS98] vorgestellten Heuristik zum Auffinden eines Covers ist die Annahme, dass die Variablen, die nach der Lösung des relaxierten Problems den höchsten LP-Lösungswert annehmen, am ehesten zu einer Verletzung des Covers führen. Ein weiterer möglicher Ansatz ist, dass die Variablen, die einen sehr fraktionellen Wert haben am ehesten zu einer Verletzung des Covers beitragen.

Um beiden Ansätzen gerecht zu werden, wird die Heuristik zweimal durchlaufen. Die beiden Durchläufe sind grundsätzlich gleich und unterscheiden sich lediglich in den unterschiedlichen Sortierreihenfolgen, der Variablen mit einem fraktionellen Wert.

Grundlage ist wieder eine Restriktion des Modells (IP) $\sum_{j\in B} a_j x_j \le b$, wobei o.B.d.A. alle Koeffizienten positiv sind.

Zur Ermittlung eines Cover Cuts werden nach und nach Variablen in ein Cover aufgenommen. Variablen werden solange aufgenommen, bis die Summe der Koeffizienten die rechte Seite übersteigt. Es sei C das gesuchte Cover, dann muss $\sum_{j\in C} a_j > b$ und

$\sum_{j\in C} a_j - a_k \le b \quad \forall k \in C$ gelten.

Da die Variablen nicht wahllos in das Cover aufgenommen werden sollen, müssen sie in eine sinnvolle Reihenfolge gebracht werden. Dazu werden sie, sortiert nach ihrem LP-Lösungswert, in Mengen eingeteilt. Sei N die Menge der Variablen, die in der betrachteten Restriktion einen Koeffizienten ungleich null haben:

$U = \left\{ j \in N \middle| x_j^* = 1 \right\}$ die Variablen mit dem LP-Lösungswert eins sind,

$L = \left\{ j \in N \middle| x_j^* = 0 \right\}$ die Variablen mit dem LP-Lösungswert null sind

und $N \setminus (U \cup L)$ die Variablen mit einem fraktionellen LP-Lösungswert darstellen.

Da die Variablen der Menge U am ehesten zu der Verletzung eines Covers beitragen wird davon ausgegangen, dass sie Teil des Covers werden. Deshalb werden sie vorerst aus der weiteren Suche herausgenommen. Folglich muss auch die rechte Seite aktualisiert werden.

$$\bar{b} = b - \sum_{j\in U} a_j$$

Um noch weitere Variablen für ein Cover zu finden, werden die Variablen der Menge $N \setminus (U \cup L)$ sortiert. Diese Sortierung ist in einem Durchlauf absteigend nach der Größe ihres aktuellen LP-Lösungswertes. In einem weiteren Durchlauf werden die Variablen nach der Fraktionalität ihres aktuellen LP-Lösungswertes sortiert. Je fraktioneller ein Wert ist, desto eher sollte die entsprechende Variable in das Cover aufgenommen werden.

Die sortierten Variablen werden nacheinander aufgenommen, bis die Summe der Koeffizienten größer als die aktualisierte rechte Seite $\bar{b}$ ist. Sei $\{k_1, k_2, \ldots, k_k\}$ die Sortierreihenfolge. Gesucht wird somit die Menge:

$K = \{k_1, k_2, ..., k_k\}$, so dass gilt $\sum_{1 \le j < k} a_{kj} \le \bar{b}$ und $\sum_{1 \le j \le k} a_{kj} > \bar{b}$

Die Mengen *U* und *K* bilden zusammen ein Cover *C*.

$$C = U \cup K$$

Beispiel:

Betrachtet man folgende Restriktion:

$$13x_1 + 15x_2 + 12x_3 + 1x_4 + 9x_5 + 11x_6 \le 20$$

mit $x^* = (0, 0.1, 1, 1, 0.5, 0)^T$

dann ergeben sich die Mengen

$U = \{3, 4\}$

$L = \{1, 6\}$

und $N \setminus (L \cup U) = \{5, 2\}$

und ein Cover

$C = \{3, 4, 5\}$

Natürlich muss das so identifizierte Cover die Bedingung $\sum_{j \in C} a_j - a_k \le b \quad \forall k \in C$ erfüllen.

Es muss geprüft werden, ob wirklich jeder beliebige Koeffizient von der Gesamtsumme abgezogen werden kann und die Summe der verbleibenden Koeffizienten immer kleiner oder gleich der rechten Seite ist. Sollte das bei einer Variablen nicht der Fall sein, muss diese Variable aus dem Cover entfernt werden. Falls diese Variable Teil der Menge *U* ist, muss ihr Koeffizient wieder zur rechten Seite $\bar{b}$ addiert werden.

Beispiel:

Bei dem gerade ermittelten, vorläufigen Cover fällt bei der Überprüfung x_4 *wieder heraus, da man* x_4 *von der Summe der Koeffizienten abziehen kann, ohne dass das Ergebnis kleiner als die rechte Seite wird.*

$$\sum_{j \in C} a_j - a_4 > a_0$$

folglich verändert sich das betrachtete Cover zu $C = \{3, 5\}$

Daraus resultiert die Ungleichung

$$x_3 + x_5 \le 1$$

Auf diese Art und Weise ergibt sich eine gültige Ungleichung, die an das Modell angehangen werden kann, ohne die ganzzahlige Lösung des Problems abzuschneiden. Ziel ist es aber, eine Ungleichung zu finden, die nicht nur gültig ist, sondern die auch die Lösung der LP-Relaxierung abschneidet. Es kann vorkommen, dass ein so ermittelter Cover Cut schon die aktuelle LP- Lösung abschneidet, wie das auch im obigen Beispiel der Fall ist. Mit Hilfe des im Folgenden vorgestellten *lifting* wird ein Cut ggf. noch verschärft und damit die Wahrscheinlichkeit erhöht, dass er verletzt ist.

4.7.4.1.2 Lifting von Cover Cuts

Liften ist der Versuch, die bisher außer Acht gelassenen Variablen der Originalrestriktion mit in einen Cover Cut aufzunehmen. Dabei muss darauf geachtet werden, dass der Cut nicht unzulässig wird. Das Grundprinzip des Liftings wurde bereits in Kapitel 4.7.3 beschrieben.

Durch das Lifting kann ein bereits gefundenes, verletztes Cover noch verschärft werden. Ein Cover wird aber auch geliftet, wenn es zunächst nicht verletzt ist, da durch das Liften die geforderte Verletzung zustande kommen kann.

Die Reihenfolge, in der die Variablen geliftet werden, ist für das Ergebnis entscheidend. Je früher eine Variable geliftet wird, umso größer ist die Wahrscheinlichkeit, dass die Variable auch wirklich in den Cut aufgenommen werden kann. Außerdem kann sie möglicherweise mit einem höheren Lifting-Koeffizienten geliftet werden. Es wird wieder auf die bereits zum Auffinden eines Covers benutzte Mengeneinteilung zurückgegriffen, die noch verfeinert wird.

Es wird folgende Aufteilung vorgenommen:

$$L = \left\{ j \in N \middle| x_j^* = 0 \right\} \text{ und } F = N \setminus (C \cup L).$$

Darüber hinaus wird das Cover C aufgeteilt in C_1 und C_2.

$$C_2 = U \text{ und } C_1 = C \setminus C_2 .$$

Als Ausgangscover wird

$$\sum_{j \in C_1} x_j \leq |C_1| - 1$$

betrachtet.

Diese Ungleichung ist eine Facette, wenn $x_j = 0, \ \ \forall j \in N \setminus C$ und $x_j = 1, \ \ \forall j \in C_2$ gilt. Können alle Variablen der Originalrestriktion geliftet werden, erhält man ebenfalls eine Facette. Um möglichst alle Variablen der Originalrestriktion liften zu können werden die Variablen der Menge C_2 vorerst aus dem Cover projiziert um dann später wieder hineingeliftet zu werden.

Das Ausgangscover muss weder von der LP-Lösung verletzt werden, noch muss es, eine gültige Ungleichung für das Originalproblem sein. Deshalb ist es zwingend erforderlich die Variablen der Menge C_2 im Laufe des Verfahrens wieder aufzunehmen. Dieser Prozess wird als *down-lifting* bezeichnet.

Beispiel:

Gegeben sei folgende Restriktion

$13x_1 + 3x_2 + 7x_3 + 6x_4 + 5x_5 + 10x_6 + 4x_7 \leq 22$

mit $x^* = (0, 0.3, 0.4, 0.7, 0.5, 1, 0.2)^T$

dann ergeben sich die Mengen

$F = \{2, 7\}$

$L = \{1\}$

$C_1 = \{5, 4, 3\}$

und $C_2 = \{6\}$

Das Ausgangscover ist in diesem Fall

$x_3 + x_4 + x_5 \leq 2$

Die Variablen der Menge F sind die einzigen Variablen, die noch Einfluss auf die Verletzung des Cover Cuts nehmen können. Sie sollen aus diesem Grunde als Erstes *up-geliftet* werden. Die Variablen der Menge L hingegen haben keinen Einfluss auf die Verletzung des Cover Cuts. Daher werden sie auch erst, nachdem die Variablen aus C_2 *down- geliftet* wurden, zu letzt betrachtet.

4.7.4.1.2.1 Up-Lifting

Das Up-Lifting dient zur Verschärfung des Cover Cuts. Ziel ist es einen Koeffizienten für eine Variable zu finden, die bisher noch nicht Bestandteil des Cover Cuts ist.

Das zu liftende Ausgangscover soll im Folgenden durch

$$\sum_{j \in C_1} \alpha_j x_j \leq \alpha_0 = |C_1| - 1$$

dargestellt werden.

w_l sei der gesuchte Lifting-Koeffizient, der sich durch $w_l = \alpha_0 - g_l$ ergibt.

$$\textit{Dabei ist } g_l = max \sum_{j \in C_1} \alpha_j x_j$$

$$\sum_{j \in C_1} a_j x_j \leq \bar{b} - a_l$$

$$x_j \in \{0,1\} \qquad \forall j \in C_1$$

g_l drückt also aus, wie viele Variablen des Cover Cuts maximal den Wert eins annehmen können, wenn die gerade betrachtete Variable x_l Teil des Covers Cuts wird und den Wert eins annimmt.

D.h. $g_l + 1$ müsste die rechte Seite sein, wenn die Variable Teil des Covers Cuts wäre. Es muss überprüft werden, ob diese erwartete rechte Seite größer ist als die des momentanen Cover Cuts. Sollte das der Fall sein, kann die Variable nicht in den Cover Cut aufgenommen werden, da unter Einbezug der zu liftenden Variablen sich eine zulässige Konstellation ergeben könnte, die dann von dem Cover Cut abgeschnitten werden würde.

Sollte die erwartete rechte Seite genauso groß sein wie die tatsächliche rechte Seite, so ist der Lifting-Koeffizient genau eins. Ist die erwartete rechte Seite kleiner, heißt das, dass sobald die zu liftende Variable den Wert eins annimmt, weniger andere Variablen den Wert eins annehmen können, als es laut der rechten Seite des Cover Cuts zu erwarten wäre. Deswegen kann der Lifting-Koeffizient dementsprechend höher sein.

Beispiel:

Wird das soeben betrachtete Problem weiterverfolgt, so muss jetzt die ermittelte Menge F geliftet werden.

Die Variable x_2 wird mit dem Koeffizienten $w_2 = 1$ geliftet

$$w_2 = \alpha_0 - g_2 = 2 - g_2$$

wobei $g_2 = max\ x_3 + x_4 + x_5$

$$7x_3 + 6x_4 + 5x_5 \leq 12 - 3$$

$$x_3, x_4, x_5 \in \{0,1\}$$

$$w_2 = 2 - 1 = 1$$

Die Variable x_7 kann nicht geliftet werden

$$w_7 = \alpha_0 - g_7 = 2 - g_7$$

wobei $g_7 = max\ x_2 + x_3 + x_4 + x_5$

$$3x_2 + 7x_3 + 6x_4 + 5x_5 \leq 12 - 4$$

$$x_2, x_3, x_4, x_5 \in \{0,1\}$$

$$w_7 = 2 - 2 = 0$$

Wird die Reihenfolge innerhalb von F allerdings vertauscht, d.h. wird x_7 vor x_2 geliftet, kann x_7 mit dem Koeffizienten 1 geliftet werden und x_2 könnte dann nicht mit in den Cover Cut aufgenommen werden.

4.7.4.1.2.2 Down-Lifting

Auch im Rahmen des Down-liftings wird ein Koeffizient für eine Variable gesucht, die bisher noch nicht Teil der Covers ist. Des Weiteren ist das Down-Lifting aber auch zwingend erforderlich, um sicherzustellen, dass die Ungleichung gültig wird. Dies ist möglich, da das Ausgangscover, mit dem der ganze Liftingprozess begonnen hat, ohne die Variablen der Menge C_2 ggf. ungültig ist.

v_l sei der gesuchte Lifting-Koeffizient, der sich durch $v_l = g_l - \alpha_0$ ergibt.

$$\textit{Dabei ist } g_l = max \sum_{j \in C_1} \alpha_j x_j$$

$$\sum_{j \in C_1} a_j x_j \leq \bar{b} + a_l$$

$$x_j \in \{0,1\} \qquad \forall j \in C_1$$

g_l drückt hier aus, wie viele Variablen den Wert eins annehmen können, wenn eine zuvor aus dem Cover Cut genommene Variable nun doch Teil des Cover Cuts wird, aber den Wert null annimmt.

Nachdem beim Herausnehmen von C_2 aus dem Cover davon ausgegangen wurde, dass die Elemente von C_2 wieder den Wert eins annehmen würden, wird hier untersucht, inwieweit sich der Cover Cut ändern würde, wenn eine Variable der Menge C_2 nicht wie erwartet den Wert eins, sondern den Wert null annehmen würde.

Können in diesem Fall weniger oder genauso viele Variablen, wie von der rechten Seite erlaubt, auf eins gesetzt werden, eröffnen sich also keine neuen Möglichkeiten durch das Aufnullsetzten der Variablen, wird diese auch nicht geliftet. Können jedoch mehr Variablen auf eins gesetzt werden, stellt die Differenz zwischen der Anzahl der Variablen, die auf eins gesetzt werden können, und der ursprünglichen rechten Seite den Lifting-Koeffizienten dar. Wären die Variablen von C_2 immer Teil des Cover Cuts geblieben, wäre ihr Koeffizient nie größer als eins.

Beim Down-lifting einer Variablen muss darauf geachtet werden, dass auch die rechte Seite der Ungleichung aktualisiert wird. Der soeben gefundene Lifting-Koeffizient wird zur rechten Seite addiert $\alpha_0 = \alpha_0 + v_l$, da die Variablen in C_2 Teil des Covers waren.

Beispiel:

Die Menge C_2 wird geliftet.

Die Variable x_6 wird mit dem Koeffizienten $v_6 = 2$ geliftet

$$v_6 = g_6 - \alpha_0 = g_6 - 2$$

$$\textit{wobei } g_6 = max\, x_2 + x_3 + x_4 + x_5$$

$$3x_2 + 7x_3 + 6x_4 + 5x_5 \leq 12 + 10$$

$$x_2, x_3, x_4, x_5 \in \{0,1\}$$

$v_6 = 4 - 2 = 2$

Nach Aktualisierung der rechten Seite sieht der Cover Cut in diesem Stadium des Liftings dann folgendermaßen aus:

$x_2 + x_3 + x_4 + x_5 + 2x_6 \leq 2 + 2$

Damit wirklich alle Variablen ggf. Bestandteil des Cover Cuts werden können, muss noch die Menge L geliftet werden.

Die Variable x_1 wird mit dem Koeffizienten $w_1 = 2$ geliftet

$w_1 = \alpha_0 - g_1 = 4 - g_1$

wobei $g_1 = max\, x_2 + x_3 + x_4 + x_5 + 2x_6$

$3x_2 + 7x_3 + 6x_4 + 5x_5 + 10x_6 \leq 22 - 13$

$x_2, x_3, x_4, x_5, x_6 \in \{0,1\}$

$w_1 = 4 - 2 = 2$

So ergibt sich der Cover Cut:

$2x_1 + x_2 + x_3 + x_4 + x_5 + 2x_6 \leq 4$

Der durch das Up- und Down-Lifting entstandene Cover Cut sieht in allgemeiner Form folgendermaßen aus:

$$\sum_{j \in C_1} \alpha_j x_j + \sum_{j \in N/C} w_j x_j + \sum_{j \in C_2} v_j x_j \leq |C_1| - 1 + \sum_{j \in C_2} v_j$$

Zur Lösung der Maximierung, die durchgeführt werden muss, um die Lifting-Koeffizienten zu ermitteln, werden Knapsack Probleme (Kapitel 3.4.3) exakt gelöst. In [GuNS98] werden die Knapsack Probleme noch nicht optimal gelöst. Es wird durch eine Heuristik eine zulässige Lösung ermittelt.

4.7.4.2 Cover Cuts in gemischt- ganzzahligen 0-1 Modellen

Bisher wurde der Anwendungsbereich von Cover Cuts erheblich dadurch eingeschränkt, dass sie nur auf Restriktionen angewandt wurden, die ausschließlich aus Binärvariablen bestehen.

Im Weiteren soll dieser Anwendungsbereich erweitert werden. Cover Cuts sollen auch von gemischten Restriktionen abgeleitet werden. Voraussetzung dafür ist, dass die Variablen, die keine Binärvariablen sind, endliche Begrenzungen haben.

Es sollen die Variablen, die keine Binärvariablen sind, aus der Restriktion entfernt werden, ohne dass sich dadurch eine unzulässige Einschränkung für die Binärvariablen ergibt. Dazu werden die nicht binären Variablen auf eine ihrer Begrenzungen gesetzt. Handelt es sich um einen positiven Koeffizienten, muss die Variable auf ihre Untergrenze gesetzt werden, um keine vorzeitige Beschränkung darzustellen. Handelt es sich um einen negativen Koeffizienten, muss die Variable auf ihre Obergrenze gesetzt werden.

Beispiel:

Gegeben sei die Restriktion

$15x_2 + 12x_3 + 11x_4 \leq 26$

wobei x_2 und x_3 Binärvariablen sind und $x_4 \in \mathbb{R}_+$ *und* $1 \leq x_4 \leq 3$

Um die Restriktion nicht einzuschränken wird x_4 auf 1 gesetzt.

Dann ergibt sich folgende entschärfte Restriktion

$15x_2 + 12x_3 + 11*1 \leq 26$ *bzw.* $15x_2 + 12x_3 \leq 15$. *Da diese Restriktion jetzt nur noch 0-1 Variablen enthält, kann das eben beschriebenen Verfahren zum Ableiten von Cover Cuts angewandt werden.*

4.7.4.3 Steuerung der Anzahl von Cover Cuts

Cover Cuts gehören zu den Klassen von Schnittebenen, deren Anzahl sich ggf. von allein beschränkt. Cover Cuts werden von den Originalrestriktionen des Modells abgeleitet. Theoretisch können alle gültigen Cover Cuts auf einmal abgeleitet werden. Dies ist möglich, wenn die Forderung, dass ein Cover Cut von der Lösung der LP-Relaxierung verletzt werden soll, aufgegeben wird und alle möglichen Reihenfolgen während des Liftings ausprobiert werden.

Bei genauerer Betrachtung des hier vorgestellten Verfahrens wird deutlich, dass dabei nicht alle grundsätzlich möglichen Cover Cuts abgeleitet werden. Z.B. haben die Variablen, die den Wert null in der relaxierten LP-Lösung annehmen, also die der Menge L, keinen Einfluss auf das Auffinden eines Covers. Genauso kann ein möglicher Cover Cut auf Grund einer bestimmten Lifting Reihenfolge nicht identifiziert werden, was auch am Beispiel in Kapitel 4.7.4.1.2 deutlich wird. Während sich, wenn x_7 vor x_5 geliftet wird, ein Cover Cut mit der Variable x_7 ergibt, wird in dem umgekehrten Fall ein Cover Cut mit der Variable x_5 abgeleitet.

Bei Anwendung des obigen Verfahrens werden allerdings auch nicht nur verletzte Cover Cuts abgeleitet. Die Cover Cuts müssen somit, bevor sie an das Modell angehangen werden, daraufhin überprüft werden, ob sie tatsächlich verletzt sind. Ansonsten kann es dazu kommen, dass in einem Durchlauf ein Cover Cut identifiziert wird, der schon an das Modell angefügt wurde. Sollen auch nicht verletzte Cover Cuts an das Modell angehangen werden, müssen die bereits vorhandenen Cuts auf mögliche Doppelungen durchsucht werden, bevor ein Cover Cut neu an das Modell angehangen wird.

Im Supernode processing vor dem Branch-and-Cut Verfahren werden Cover Cuts in Runden abgeleitet. In einer Runde werden alle Originalrestriktionen auf der Suche nach Cover Cuts durchlaufen. Anschließend wird reoptimiert, bevor in der nächsten Runde die Suche von neuem beginnt. Das gesamte Verfahren braucht im Prinzip keine explizite Abbruchbedingung. Cover Cuts können theoretisch so lange abgeleitet werden, bis keine mehr gefunden werden. Dennoch hat es sich als sinnvoll erwiesen, eine Abbruchsbedingung einzuführen. Gemäß dieser Bedingung soll die Suche nach Cover Cuts enden, wenn die Veränderung des Zielfunktionswertes einen bestimmten Wert unterschreitet. Da die Vorgehensweise in Runden aufgeteilt ist, besteht eine weitere Abbruchsbedingung darin, die Anzahl der Runden zu beschränken.

Sollen Cover Cuts an einem ausgewählten Knoten im Baum abgeleitet werden, wird nur eine Runde nach ihnen gesucht.

4.7.5 Mixed Integer Gomory Cuts

Bereits Ende der fünfziger bzw. Anfang der sechziger Jahre wurde der fraktionelle Gomory Cut [Gomo58] und dem folgend der darauf aufbauende Mixed Integer Gomory Cut [Gomo60; GiGo63] als Lösungsansatz für ganzzahlige Probleme vorgestellt. Diese Cuts zeichnen sich vor allem dadurch aus, dass immer ein verletzter Cut gefunden werden kann, wenn eine Variable, für die Ganzzahligkeit gefordert ist, fraktionell ist. Sie reichen zur vollständigen polyedrischen Beschreibung der konvexen Hülle der Lösungsmenge aus. Der Ansatz ein ganzzahliges Problem allein mit Hilfe dieser Gomory Cuts zulösen, kam jedoch bei vielen Modellen an seine Grenzen und wurde bald wieder verworfen. In Verbindung mit einem Branch-and-Cut Ansatz wurde der Mixed Integer Gomory Cut wieder aufgegriffen und kann bei vielen Modellen zu einer erheblichen Verbesserung der Lösungszeiten führen (vgl. [BCCN96]).

Die Basisvariable x_{Bk} wird im optimalen Simplextableau wie folgt dargestellt:

$$x_{Bk} + \sum_{j \in NB_I} \bar{a}_{kj} x_j + \sum_{j \in NB_C} \bar{a}_{kj} x_j = \bar{b}_k$$

wobei NB_I und NB_C die Menge der ganzzahligen bzw. kontinuierlichen Nicht-Basisvariablen beschreibt.

Von jeder Zeile des Tableaus der optimalen LP-Lösung, in der eine ganzzahlige Variable fraktionell und in der Basis ist, kann ein Cut abgeleitet werden, der von der optimalen Lösung der Relaxierung verletzt wird.

Ist $\bar{b}_k \notin \mathbb{Z}_+$ und $\bar{b}_k = \lfloor \bar{b}_k \rfloor + \langle \bar{b}_k \rangle$, $\bar{a}_{kj} = \lfloor \bar{a}_{kj} \rfloor + \langle \bar{a}_{kj} \rangle$ lässt sich der Mixed Integer Gomory Cut wie folgt beschreiben:

$$\sum_{j \in NB_I^1} \langle \bar{a}_{kj} \rangle x_j + \sum_{j \in NB_C^+} \bar{a}_{kj} x_j + \sum_{j \in NB_I^2} \frac{\langle \bar{b}_k \rangle}{(1 - \langle \bar{b}_k \rangle)} (1 - \langle \bar{a}_{kj} \rangle) x_j + \sum_{j \in NB_C^-} \frac{\langle \bar{b}_k \rangle}{(1 - \langle \bar{b}_k \rangle)} \bar{a}_{kj} x_j \geq \langle \bar{b}_k \rangle \tag{4.5}$$

Dabei gilt:

$$NB_I^1 = \{ j \in NB_I \mid \langle \bar{a}_{kj} \rangle \leq \langle \bar{b}_k \rangle \}$$

$$NB_I^2 = \{ j \in NB_I \mid \langle \bar{a}_{kj} \rangle > \langle \bar{b}_k \rangle \}$$

$$NB_C^+ = \{ j \in NB_C \mid \langle \bar{a}_{kj} \rangle > 0 \}$$

$$NB_C^- = \{ j \in NB_C \mid \langle \bar{a}_{kj} \rangle \leq 0 \}$$

Die Herleitung des Mixed Integer Gomory Cuts ist beispielsweise in [Wols98] zu finden.

Beispiel:

Der Mixed Integer Gomory Cut soll an folgendem Modell erläutert werden:

$$max\ 4x_1 - x_2$$

$$7x_1 - 2x_2 \leq 14$$

$$x_2 \leq 3$$

$$2x_1 - 2x_2 \leq 3$$

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

Das Simplex-Tableau der Relaxierten Lösung sieht dazu folgendermaßen aus:

x_1	x_2	x_3	x_4	x_5	
		-4/7	-1/7		59/7
1	0	1/7	2/7	0	20/7
0	1	0	1	0	3
0	0	-2/7	10/7	1	23/7

Aus der ersten Zeile lässt sich der Gomory Cut

$$\frac{1}{7}x_3 + \frac{2}{7}x_4 \geq \frac{6}{7} \ \textit{bzw.}\ x_1 \leq 2,$$

ableiten. Wird das Modell um diesen Cut erweitert und reoptimiert, ergibt sich $x^* = (2;0,5;1;2,5;0)$. Da lediglich für x_1 Ganzzahligkeit gefordert ist, ist mit dieser Lösung auch das Ausgangsmodell gelöst.

Der Mixed Integer Gomory Cut wurde implementiert. Wenn im weiteren Verlauf der Arbeit vereinfacht vom Gomory Cut die Rede ist, ist dieser soeben vorgestellte Mixed Integer Gomory Cut gemeint.

4.7.5.1 Steuerung der Anzahl von Gomory Cuts

Aufgrund der Tatsache, dass durch endliches Anwenden der Mixed Integer Gomory Cuts theoretisch die optimale ganzzahlige Lösung gefunden werden kann, gewinnt die Frage nach der sinnvollen Anzahl an Cuts sehr große Bedeutung. Wie bei allen Schnittebenen ist die Erweiterung des Modells um Restriktionen, mit der damit verbundenen Erschwernis bei der LP-Optimierung, gegen die durch die Schnittebenen hervorgerufene Einschränkung des Lösungsraums für die ganzzahlige Lösung abzuwägen. Insbesondere Gomory Cuts können die Geschwindigkeit der LP-Optimierung deutlich beeinträchtigen, da es sich in der Regel um dicht besetzte Cuts handelt.

Im Folgenden sollen mehrere Möglichkeiten diskutiert werden, wie und wann die Suche nach Gomory Cuts eingeschränkt werden kann.

Von einer Basis kann *ein* Gomory Cut abgeleitet werden. Dieser verletzt die optimale Lösung, so dass sich nach einer erneuten Optimierung eine neue Basis ergibt. von dieser neuen Basis kann dann wieder ein Gomory Cut abgeleitet werden. Gleichermaßen ist es auch möglich, mehrere Gomory Cuts von einer Basis abzuleiten ggf. sogar für jede fraktionelle Variable in einer Basis, für die Ganzzahligkeit gefordert ist. Die Arbeit [BCCN96] beschäftigt sich mit der Thematik, wie viele Gomory Cuts von einer Basis abgeleitet werden sollten. Im Ergebnis wird festgestellt, dass am besten von allen fraktionellen Variablen ein Gomory Cut abgeleitet werden sollte.

Ein weiterer Ansatzpunkt ist, nur die fraktionellen Variablen zu wählen, die einen bestimmten Grad an Fraktionalität aufweisen. Ist eine Variable schon sehr nahe an der Ganzzahligkeit, wird sie nicht mehr als Grundlage für das Ableiten eines Cuts verwendet. Mit dieser Strategie beschränkt sich die Anzahl der Cuts, die von einer Basis abgeleitet werden ggf. von ganz allein.

Die bisher in Betracht gezogenen Strategien hatten das Ziel, weniger Gomory Cuts zu generieren. Mitunter ist es aber auch sinnvoll, einen bereits generierten Cut zu verwerfen, anstatt ihn an das Modell anzuhängen.

Ein Bewertungskriterium, in dem sich die Cuts unterscheiden, ist analog zu allen anderen Cuts, auch bei den Gomory Cuts der Grad der Verletzung. Nur Cuts die einen bestimmen Grad an Verletzung aufweisen, sollten an das Modell angehangen werden.

Darüber hinaus sollten Gomory Cuts, die besonders dicht besetzt sind, aussortiert werden. Sehr dicht besetze Cuts können die folgenden LP-Optimierungen erheblich verlangsamen.

Sowohl die Abfrage nach dem Grad der Verletzung als auch die Abfrage nach der Anzahl der Elemente in einem Cut sind trivial.

Die Behauptung, dass ein Gomory Cuts mit einer großen Anzahl von Nicht-Null-Elementen die LP-Lösungszeit beeinträchtigt, muss etwas genauer betrachtet werden. Es wurde davon ausgegangen, dass sich die LP-Lösungszeit verlangsamt, weil ein dicht besetzter Gomory Cut dazu führt, dass in der LU-Faktorisierung der modifizierten Basismatrix mehr Nicht-Null-Elemente produziert werden. Diese Annahme ist nahe liegend jedoch nicht zwingend. Ein weniger dicht besetzter Gomory Cut kann theoretisch für eine größere Auffüllung sorgen als ein sehr dicht besetzter Gomory Cuts.

Zur Identifikation der Gomory Cuts, die zu besonders vielen, neuen Nicht-Null-Elementen führen, wurde eine Abwandlung des Markowitz Kriteriums [Mark57] zu Hilfe genommen werden.

In MOPS wird das Markowitz Kriterium bereits im Rahmen der LU-Faktorisierung verwendet. Dabei wird es bei der Auswahl eines Pivotelements berücksichtigt. Auch in dieser Situation kommt die Motivation, das Markowitz Kriterium einzusetzen daher, dass ein Pivotelement gefunden werden soll, welches in der folgenden LU-Faktorisierung möglichst wenig neue Nicht-Null-Elemente erzeugt.

Mit einer Modifikation dieses Markowitz Kriteriums [SuFW07] soll die Auffüllung prognostiziert werden, die ein Gomory Cut produzieren kann, wenn ein Element des Cuts als Pivotelement ausgewählt wird. Im Vergleich zu der ursprünglichen Anwendung des Markowitz Kriteriums sollen im Rahmen der Gomory Cuts nicht einzelne Elemente sondern ganze Zeilen identifiziert werden. Es wird also das Markowitz Kriterium für alle Variablen des Gomory Cuts G, die einen Koeffizienten ungleich null haben, aufsummiert. Sei $G \subseteq NB$ die Menge der Nicht-Null-Elemente im Gomory Cut und n_j die Anzahl der Nicht-Null-Elemente in der Spalte j, dann ist $MK = \sum_{j \in G} (n_j - 1) / |G|$ die durchschnittliche Anzahl von Nicht-Null-Elementen in den Spalten, die zu den Nicht-Null-Elementen des Gomory Cuts gehören.

Die Cuts, die die kleinsten Werte für MK aufweisen sollten an das Modell angehangen werden. Da aber das erneute Generieren der Gomory Cuts oder das Zwischenspeichern aller Gomory Cuts zu aufwendig ist, werden die Werte der Cuts für MK nicht miteinander, sondern mit einer vorher festgesetzten Obergrenze (in MOPS ein Vielfaches der Anzahl der Variablen des Modells) verglichen.

4.7.6 Clique Cuts

4.7.6.1 Allgemeine Clique Cuts

Das hier beschriebene Verfahren zum Ableiten von Clique Cuts basiert auf den Arbeiten von [Save94] und [SuSz94].

Es soll folgende Restriktion aus 0-1-Variablen betrachtet werden.

$$\sum_{j \in B} a_j x_j \le b$$

$$x_j \in \{0,1\}, \qquad \forall j \in B$$

Die Menge $C \subseteq B$ definiert eine Clique, wenn gilt

1.) $\left(j \in C^+ \wedge x_j = 1\right) \rightarrow \left(\forall k \in C^+ - \{j\} \rightarrow x_k = 0 \wedge \left(\forall k \in C^- \rightarrow x_k = 1\right)\right)$

2.) $\left(j \in C^- \wedge x_j = 0\right) \rightarrow \left(\forall k \in C^- - \{j\} \rightarrow x_k = 1 \wedge \left(\forall k \in C^+ \rightarrow x_k = 0\right)\right)$

3.) $|C| \ge 2$. (4.6)

Aus graphentheoretischer Sicht definiert sich eine Clique folgendermaßen. Sei G ein ungerichteter Graph mit der Knotenmenge B, der Kantenmenge E, ohne Mehrfachkanten und C eine Teilmenge von B, dann wird C als Clique von G bezeichnet, wenn für je zwei beliebige verschiedene Knoten aus C gilt, dass sie durch eine Kante miteinander verbunden sind.

Auch die betrachtete Restriktion $\sum_{j \in B} a_j x_j \le b$ lässt sich als Graph darstellen. Die Knoten repräsentieren die einzelnen 0-1- Variablen und eine Kante zwischen zwei Knoten drückt aus, dass die zwei verbundenen 0-1-Variablen nicht beide den Wert eins annehmen dürfen.

Eine Restriktion kann mehrere Cliquen beinhalten.

Wurde eine Clique erstmals identifiziert, kann darauf basierend, wie z. B. in [NeWo88] gezeigt wurde der folgende Cut abgeleitet werden.

$$\sum_{j \in C^+} x_j - \sum_{j \in C^-} x_j \le 1 - \left|C^-\right|$$

Beispiel:

Aus der Restriktion

$$15x_1 + 13x_2 - 10x_3 - 10x_4 + 7x_5 + 5x_6 \le -1$$

gehen folgende Cliquen hervor $C_1 = \{1, 2, 3, 4\}$; $C_2 = \{1, 2, 5\}$; $C_3 = \{1, 6\}$.

Abbildung 4.9 stellt diese Cliquen grafisch dar.

Die passenden Clique Cuts dazu sind dann

$$x_1 + x_2 - x_3 - x_4 \le 1 - 2$$

$$x_1 + x_2 + x_5 \le 1 \text{ und } x_1 + x_6 \le 1$$

Wenn einer dieser Cuts von der Lösung der LP-Relaxierung verletzt wird, wird er an das Modell angehangen.

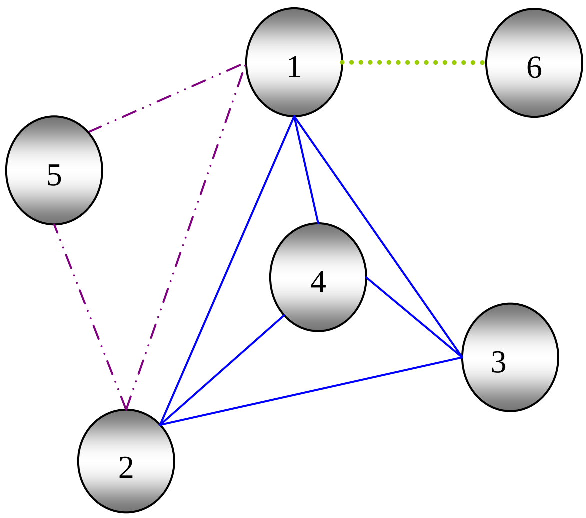

Abbildung 4.9: Darstellung von Cliquen

Die Restriktionen von denen Cliquen identifiziert werden, müssen nicht reine 0-1-Restriktionen sein. Kontinuierliche oder allgemeine Integervariablen werden, wenn sie einen positiven Koeffizienten haben, auf ihre Untergrenze, und wenn sie einen negativen Koeffizienten haben, auf ihre Obergrenze gesetzt. Sie können allerdings nicht Teil einer Clique sein.

Clique Cuts sind relativ leicht abzuleiten, insbesondere wenn die Koeffizienten der 0-1-Variablen einer Restriktion in absteigender Reihenfolge gespeichert sind.

4.7.6.2 Erweiterte Clique Cuts

Das im vorangegangenen Abschnitt beschriebene Verfahren zur Identifikation von Clique Cuts basiert auf einer Zeile des Modells. Um schärfere Clique Cuts identifizieren zu können, müssen alle 0-1-Variablen in einem Modell in Betracht gezogen werden. Die Definition einer Clique (4.6), wie auch schon in [Save94] und [DiEC93]gezeigt wurde, ist allgemein und nicht auf eine Zeile beschränkt.

Es sollen die folgenden zwei Restriktionen betrachtet werden:

$$15x_1 + 13x_2 + 10x_3 + 10x_4 + 7x_5 + 5x_6 \leq 19$$

$$3x_2 + 5x_3 + 5x_4 + 6x_5 + 8x_6 \leq 10$$

Alle aus diesen beiden Restriktionen hervorgehenden Cliquen sind grafisch dargestellt in Abbildung 4.10. Aus der ersten Zeile geht u. a. die Clique $C = \{1, 6\}$ *hervor. Unter Einbeziehung der zweiten Restriktion kann diese Clique zu der Clique* $C = \{1, 2, 3, 4, 5, 6\}$ *erweitert werden. Der aus dieser Clique hervorgehende Clique Cut* $x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \leq 1$ *ist nicht nur schärfer als der der Ausgangsclique sondern dominiert auch alle anderen Clique Cuts der Cliquen, die von den beiden Restriktionen abgeleitet werden können.*

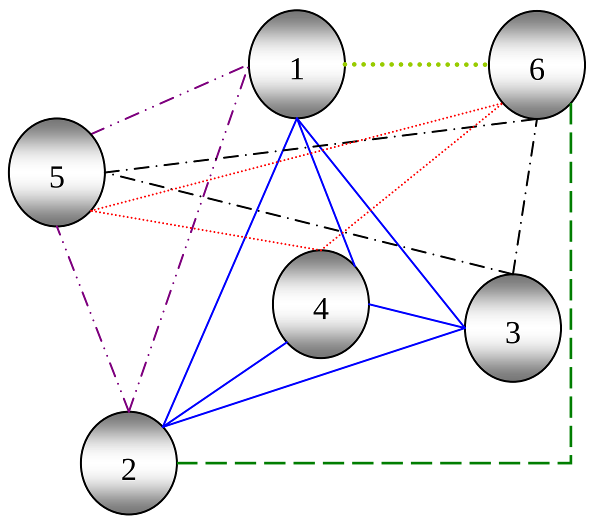

Abbildung 4.10: Darstellung von Cliquen aus zwei Restriktionen

Die in dieser Arbeit angestellten Überlegungen basierten auf einem 0-1-Ungleichungssystem. Dieses Ungleichungssystem kann entweder aus einfachen Cliquen bestehen, die von jeweils einer Zeile des Modells abgeleitet wurden oder es handelt sich bei dem zu lösenden Modell um ein Set Partitioning oder Set Packing Problem (Kapitel 3.4.2). In [HoPa93; KrFP03] werden weitere Preprocessing-Techniken, die nur spezielle für diese Art von Problemen anwendbar sind, beschrieben.

Um das hier vorgestellte Verfahren einsetzen zu können, reicht es sogar aus, wenn Teile des Modells ein 0-1-Ungleichungssystem darstellen:

$$
\begin{aligned}
& min \quad \sum_{j \in B} c_j x_j \\
& \qquad\quad Ax \le d \\
& mit \quad \{x \mid Tx \le e\} \subseteq \{x \mid Ax \le d\} \\
& wobei\ T \in \{0,1\}^{pxn}\ und\ e \in \{1\}^p,\ p \le m. \\
& \qquad\quad x_j \in \{0,1\} \qquad\qquad \forall j \in B \qquad\qquad\qquad \text{(IP)}
\end{aligned}
$$

Es sei I die Indexmenge der Restriktionen und J die Indexmenge der Variablen, die Teil des 0-1 Ungleichungssystems sind.

Lemma: $\sum_{j \in J_i} x_j \le 1$ sei eine beliebige Clique-Restriktion mit $i \in I$ und $J_i \subset J$. Es sei $s \in J - J_i$ und gelte $\langle a_s ; a_j \rangle \ne 0 \quad \forall j \in J_i$.

Dann ist $\sum_{j \in J_i} x_j + x_s \le 1$ eine gültige Ungleichung für (IP).

Beweis: Sei x^* eine beliebige zulässige Integer-Lösung für (IP).

Fall1: $\sum_{j \in J_i} x_j^* = 0$, daraus folgt $\sum_{j \in J_i} x_j^* + x_s^* \le 1$.

Fall2: $\sum_{j \in J_i} x_j^* = 1$, d.h. $\exists k \in J_i$ mit $x_k^* = 1$, da $\langle a_s ; a_k \rangle \ne 0$ muss $x_s^* = 0$ sein. Also gilt $\sum_{j \in J_i} x_j^* + x_s^* \le 1$.

Entscheidend für eine gültige Erweiterung einer Clique Restriktion ist, dass alle Variablen, die in eine Clique Restriktion aufgenommen werden sollen, in dem 0-1-Ungleichungssystem nicht orthogonal zu allen, sich in der Clique Restriktion befindenden Variablen sind.

Ziel ist es durch einen effizienten Lifting-Prozess gegebene Clique Restriktionen zu verschärfen. Die Arbeit [DEGP93] hingegen beschäftigt sich damit, möglichst maximale Cliquen auf zu finden. Dabei handelt es sich allerdings um ein NP-schweres Problem. Eine Übersicht der Literatur zum *maximum clique problem* ist in [PaXu94] zu finden.

Im folgenden soll ein Verfahren beschrieben werden, dass dazu dient gegebene Clique Restriktionen zu erweitern. Abbildung 4.11 zeigt den Ablauf des Erweiterns einer Clique.

Zu Beginn des Verfahrens wird eine Mengeneinteilung vorgenommen. Dabei sei $L = \{j \,|\, j \in J \wedge x_j^{\,*} = 0\}$ und $K = J \setminus L$. Dem folgend wird eine Clique gewählt. Sei $\sum_{j \in J_i} x_j \le 1$ eine beliebige Clique *i* mit $i \in I$ und $J_i \subset J$. Aus dieser Clique geht die Ausgangsclique C_i hervor $C_i \subset J_i$. Teil dieser Ausgangsclique sind alle Variablen der Clique *i,* die in der LP-Relaxierung des IP-Modells nicht den Lösungswert null haben, da nur sie zu einer Verletzung beitragen können. Die Ausgangsclique C_i soll im Folgenden erweitert werden. Wenn eine zusätzliche Variable x_s mit $s \notin C_i$ Teil dieser Ausgangsclique werden soll muss sie folgende Bedingung erfüllen:

$$\langle a_s ; a_j \rangle \neq 0 \qquad \forall j \in C_i \tag{4.7}$$

Wobei *a* der Koeffizientenvektor einer Variablen im 0-1-Ungleichungssystem ist. Der Koeffizientenvektor der Variablen x_s muss also daraufhin überprüft werden, ob er nicht orthogonal zu den Koeffizientenvektoren aller Variablen aus C_i ist. Trifft das für die Variable x_s zu kann sie in C_i aufgenommen werden. Es können nur Variablen Teil der Clique werden, die mit jeder Variablen aus C_i in mindestens einer Zeile zusammen auftreten. Die Reihenfolge, in der zusätzliche Variablen in die Ausgangsclique aufgenommen werden ist von großer Bedeutung. Jede Variable muss für alle, also auch für die vor ihr neu aufgenommen Variablen Bedingung (4.7) erfüllen. Aus diesem Grund sollen nur die Variablen mit einem fraktionellen Lösungswert betrachtet werden. Nur die Variablen der Menge $S_i = K \setminus C_i$ sollen, wenn möglich, in die Ausgangsclique aufgenommen werden.

Es wird davon ausgegangen, dass es immer Variablen gibt, die in der LP-Relaxierung einem Wert ungleich null haben und nicht Teil der Ausgangsclique sind $C_i \subset K$.

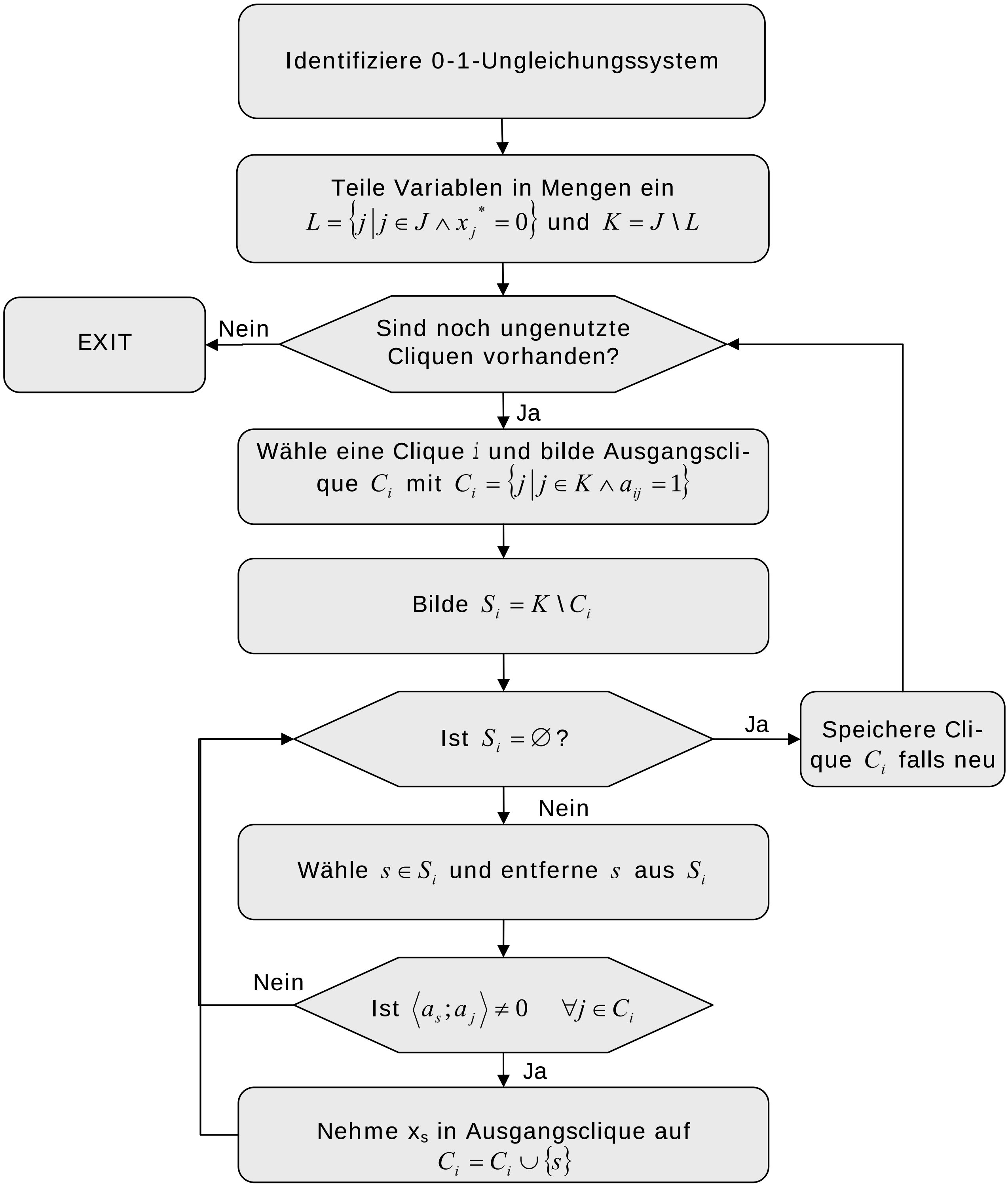

Abbildung 4.11: Vorgehen zur Erweiterung einer Clique

Beispiel:

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9		RHS
	1			1	1			1	1	=	1
	1	1	1							=	1
	1			1	1		1			=	1
					1	1	1		1	=	1
$x^*=$	(0.5;	0.0;	0.5;	0.3;	0.0;	0.6;	0.2;	0.0;	0.2)		

$K = \{1,3,4,6,7,9\}$

$C_1 = \{1, 4, 9\}$ *und* $S_1 = \{3, 6, 7\}$

Die Variablen aus S_1 werden nacheinander daraufhin überprüft, ob sie nicht orthogonal zu allen Variablen aus C_1 sind. Da $\langle a_3; a_4 \rangle = 0$ und $\langle a_3; a_9 \rangle = 0$ kann x_3 nicht in C_1 aufgenommen werden. Genauso wie x_6, da $\langle a_6; a_1 \rangle = 0$ und $\langle a_6; a_4 \rangle = 0$. Nur für die Variable x_7 gilt $\langle a_7; a_j \rangle \neq 0 \quad \forall j \in C_i$ folglich kann x_7 in C_1 aufgenommen werden. Mit $x_1 + x_4 + x_7 + x_9 \leq 1$ wird so eine neue und verletze Clique gefunden.

Basiert ein neuer Clique Cut nicht auf einem aus Clique Cuts bestehenden Ungleichungssystem sondern auf Restriktionen, die schon Teil des Modells sind, kann die getroffene Mengeneinteilung verfeinert werden zu:

$$L = \{j \mid j \in J \wedge x_j^* = 0\},\ U = \{j \mid j \in J \wedge x_j^* = 1\} \text{ und } K = J \setminus (L \cup U)$$

Denn die Variablen mit einem Wert von eins tragen auch nicht zur Verletzung bei, da alle *benachbarten* Knoten (Variablen) den Wert null haben müssen. Mit benachbarten Knoten, sind alle Variablen gemeint, die in einem Clique-Graph mit einer Variablen verbunden sind, d.h. alle Variablen die zusammen mit der Variablen in einer Zeile zu finden sind.

Das beschriebene Verfahren wurde im Rahmen dieser Arbeit als Prototyp für Set Partitioning oder Set Packing Probleme implementiert. In anschließenden Arbeiten, soll eine weiterreichende Implementierung und vollständige Integrierung in das MOPS System erfolgen.

4.7.7 Implication Cuts

4.7.7.1 Das Probing

Wie ihr Name schon sagt, basieren Implication Cuts auf im Modell gegebenen Implikationen, welche durch ein Probing identifiziert werden. Während des Probings wird eine beliebige noch nicht fixierte 0-1-Variable versuchsweise auf 0 bzw. 1 gesetzt. Diese Setzung kann dazu führen, dass Restriktionen unzulässig werden oder andere noch nicht fixierte Variablen fixiert werden können. Wenn eine neue Fixierung eintritt, wird untersucht, inwieweit diese auf andere noch freie Variablen Einfluss hat, d.h. ob wiederum weitere Variablen fixiert werden können. Im Laufe des Probings werden also die Implikationen auf andere Variablen, die ein Setzen einer 0-1-Variablen hervorruft, festgestellt. Dabei können nicht nur 0-1-Variablen fixiert werden, sondern auch allgemeine, ganzzahlige und kontinuierliche Variablen.

Neben der Tatsache, dass es die Grundlage für das Ableiten von Implication Cuts darstellt, ist das Probing noch aus folgenden anderen Gründen von Bedeutung:

- Variablen können global fixiert werden, wenn Widersprüche auftreten.
- Unlösbare Probleme können identifiziert werden.
- Die Implikationen können benutzt werden, um Schranken von kontinuierlichen Variablen zu verschärfen.
- Die Implikationen können während des Branch-and-Bound/Cut Verfahrens für die Auswahl der Branching-Variable benutzt werden.
- Das Probing und die daraus hervorgehenden Implikationen werden als Grundlage für die erweiterte Koeffizientenreduktion benötigt [SuSu99].

Im Folgenden soll kurz an einem Beispiel das Prinzip der Implication Cuts dargestellt werden. Alle weiteren aus dem Probing hervorgehenden Möglichkeiten und eine detaillierte Beschreibung der Implication Cuts sind zu finden in [SuSz94].

Beispiel:

	x_1	x_2	x_3	x_4	x_5	x_6	*Typ*	*RHS*
min	*7*	*6*	*3*	*2*	*4*	*2*		
R1	*-2*	*-2*			*4*		$\leq$	*0*
R2				*1*	*5*	*2*	$\geq$	*25*
R3		*-2*				*1*	$\leq$	*0*
R4	*-12*		*1*				$\leq$	*0*
R5			*6*	*1*			$\geq$	*30*
LP-sol.	*0,07*	*0,0*	*0,86*	*24,82*	*0,0*	*0,0*		

Dabei sind $x_1, x_2, x_3 \in \{0,1\}$ *und* $0 \leq x_4, x_5, x_6 \leq 30$.

Wenn $x_1=0$ *dann muss wegen R4 auch* $x_3 = 0$ *sein. Wenn* $x_3 = 0$ *ist muss in R5* x_4 *die Restriktion alleine erfüllen, d.h.* $x_4 = 30$.

Insgesamt sind in diesem Modell folgende Implikationen zu finden:

$x_1 = 0 \quad \Rightarrow \quad x_5 = 0; x_3 = 0; x_4 = 30$

$x_2 = 0 \quad \Rightarrow \quad x_5 = 0; x_6 = 0; x_4 \geq 25$

$x_3 = 0 \quad \Rightarrow \quad x_4 = 30$

Allgemein lassen sich aus den ermittelten Implikationen, folgende Ungleichungen entwickeln:

$y = 1 \Rightarrow x = a$	$x \geq l + (a-l)y$ $x \leq u - (u-a)y$
$y = 0 \Rightarrow x = a$	$x \geq a - (a-l)y$ $x \leq a + (u-a)y$

Wobei $y \in \{0,1\}$ *und* $l \leq x \leq u$

Schneidet eine dieser Ungleichungen die Lösung der aktuellen LP-Relaxierung ab, wird sie als Implikation Cut an das Originalmodell angehangen, so dass für das obige Beispiel der folgende Implication Cut abgeleitet werden kann:

$$x_4 \geq 30 - 30x_1$$