5 Branch-and-Bound Verfahren

5.1 Grundstruktur des Branch-and-Bound Verfahrens

Das Branch-and-Bound Verfahren ist ein exaktes Verfahren für diskrete Optimierungsprobleme. Es wurde in den 60ziger Jahren vorgestellt [LaDo60; Daki65]. Durch vielfältige Untersuchungen konnte eine stetige Verbesserung erzielt werden. Heute basieren fast alle Optimierungssysteme zur Lösung von gemischt-ganzzahligen Modellen auf dem Branch-and-Bound bzw. dem im Anschluss vorgestellten Branch-and-Cut Verfahren [Four05].

Im Rahmen des Verfahrens wird zum Beweis einer optimalen Lösung, deren Existenz vorausgesetzt, der gesamte Lösungsraum durchsucht. Ausgangspunkt für das Branch-and-Bound Verfahren ist die LP-Relaxierung. Der zulässige Bereich des Ausgangsproblems wird in zwei disjunkte Teilmengen zerlegt. Damit erfolgt eine Verzweigung des Ausgangsmodells in Teilprobleme (branching).

Während des Branch-and-Bound Verfahrens können verschieden Arten von ganzzahligen Variablen und verschiedenen Gruppen von Variablen beachtet werden (vgl. [BeFo78; BeFo79; BeTo70; Frie07]). Dazu zählen:

- Allgemeine Integer Variablen.
- Special Ordered Sets: Eine Gruppe von Variablen, die alle 0-1 Variablen sind und in der Summe 1 ergeben sollen.
- Special Ordered Sets vom Typ 1: Eine Gruppe von Variablen, bei denen höchstens eine Variable nicht Null sein darf.
- Special Ordered Sets vom Typ 2: Eine Gruppe von Variablen, bei denen höchstens zwei, nebeneinander liegende Variablen nicht Null sein dürfen.
- Partial Integer Variablen: Variablen, die nur in einem bestimmten Bereich ganzzahlig sein müssen.
- Semi Continuous Variablen: Variablen, die einen Wert in dem Bereich $\{0\} \cup [l,u]$ annehmen müssen. Wobei $0 < l < u$ und $l,u \in \mathbb{R}$.

- Semi Integer Variablen: Variablen, die einen Wert in dem Bereich $\{0\} \cup [l,u]$ annehmen müssen. Wobei $1 \le l < u$ und $l,u \in \mathbb{Z}$.

Im Weiteren soll das Prinzip des Branch-and-Bound Verfahrens an dem einfachen Fall von allgemeinen Integer Variablen verdeutlicht werden.

Betrachtet wird folgendes Modell:

$$\begin{aligned} min \quad & \sum_{j \in B} c_j x_j \\ & Ax \le d \qquad (P_0) \\ & x_j \in \mathbb{Z}_+ \qquad \forall j \in I \end{aligned}$$

Dieses Modell wird relaxiert gelöst. Es wird eine Variable x_j ausgewählt, für die Ganzzahligkeit gefordert ist, welche aber bei der Lösung der LP-Relaxierung einen fraktionellen Wert x_j^* annimmt. Auf der Grundlage des Ausgangsproblems (P_0) werden zwei Teilprobleme entwickelt, indem einmal die Variable nach unten beschränkt wird $x_j \ge \lfloor x_j^* \rfloor + 1$ und das andere Mal nach oben $x_j \le \lfloor x_j^* \rfloor$. So entstehen die Probleme (P_1) und (P_2). Der Bereich in dem die Variable x_j einen Wert zwischen $\lfloor x_j^* \rfloor$ und $\lfloor x_j^* \rfloor + 1$ annimmt, wird ausgeschlossen. Alternativ zur Einführung von zusätzlichen Restriktionen, die die Beschränkungen bewirken kann auch die Obergrenze ub_j bzw. Untergrenze lb_j der Variablen x_j angepasst werden. Die optimale ganzzahlige Lösung muss sich in einem der beiden Teilprobleme befinden. Die Teilprobleme werden relaxiert gelöst. Jedes der beiden Teilprobleme wird anhand einer neu gewählten Variablen, der Branching-Variablen, ggf. wieder in zwei Teilprobleme zerlegt. Dieser Prozess kann durch einen Baum (s. Abbildung 5.1) visualisiert werden.

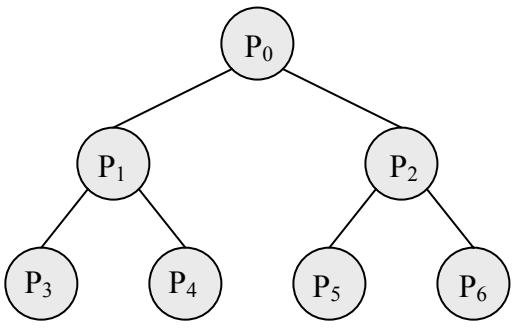

Abbildung 5.1: Ein Enumerationsbaum

Während das Ausgangsproblem durch den Wurzelknoten des Baums repräsentiert wird, stellt jeder weitere Knoten ein erzeugtes Teilproblem dar. Die Kanten zwischen den Knoten symbolisieren die Beschränkungen der Branching-Variablen, durch die der zulässige Bereich eines jeden relaxierten Teilproblems verkleinert wird. Somit stellt jeder Knoten entlang eines Pfades ein Teilproblem dar, das durch die Menge der Restriktionen entlang des Pfades bis zum Wurzelknoten beschränkt wird. Je tiefer sich ein Knoten im Enumerationsbaum befindet, desto kleiner ist der zulässige Bereich und damit die Menge möglicher Lösungen des korrespondierenden relaxierten Teilproblems [GaNe72].

Ein durch einen Knoten repräsentiertes Teilproblem, lasst sich allgemein folgendermaßen darstellen:

$$
\begin{aligned}
min \quad & \sum_{j \in I} c_j x_j \\
& \overline{A}x \le \overline{d} \\
& x_j \in \mathbb{Z}_+ \qquad\qquad \forall j \in I
\end{aligned}
$$

wobei $\overline{A}x \le \overline{d}$, die Originalrestriktionen $Ax \le d$ erweitert um die Beschränkungen $x_j \le \lfloor x_j^* \rfloor, j \in F_u$ und $x_j \ge \lfloor x_j^* \rfloor + 1, j \in F_l$ ist. F_u und F_l stellen die Mengen der Branching-Variablen, die nach oben bzw. nach unten beschränkt wurden dar.

Wenn also $F_u = \varnothing$ und $F_l = \varnothing$ entspricht das obige Modell dem Originalmodell.

An jedem Knoten wird ein Teilproblem relaxiert gelöst. Dieses Teilproblem stellt das um die Beschränkungen erweiterte Originalproblem dar. Aufgrund der mit der Tiefe des Baumes steigenden Komplexität der Teilprobleme ist ein möglichst kleiner Baum wünschenswert.

Während der Abarbeitung der Teilprobleme ergeben sich für den Bereich, in dem die optimale ganzzahlige Lösung liegen kann, neue Ober- und Untergrenzen. Da der Zielfunktionswert der LP-Relaxierung am ersten Knoten mindestens so gut wie der der IP-Lösung ist, stellt er vorerst eine globale Untergrenze für den Zielfunktionswert des IP-Problems dar. In Kapitel 4 wurde dargestellt, wie während des Supernode processings diese Untergrenze als Ausgangsposition für das Branch-and-Bound Verfahren so weit wie möglich heraufgesetzt wird. Innerhalb des Baumes wird die globale Untergrenze immer auf den Wert gesetzt, der minimal unter den Zielfunktionswerten der LP-Relaxierungen der noch nicht abgearbeiteten Knoten ist.

Eine globale Obergrenze ist nur dann gegeben, wenn eine gültige IP-Lösung bekannt ist. Am Ausgangsknoten könnte diese durch Heuristiken gefunden worden sein. Alternativ kann durch den Anwender eine auf Erfahrungen basierende IP-Lösung angegeben werden. Steht zu Beginn des Branch-and-Bound Verfahrens keine gültige IP-Lösung zur Verfügung, wird die globale Obergrenze auf $+\infty$ gesetzt. Sobald eine gültige ganzzahlige Lösung gefunden wird, wird die globale Obergrenze durch diese ersetzt. Damit stellt die globale Obergrenze immer gleichzeitig die aktuell beste bekannte IP-Lösung dar.

Ziel ist es, die globale Untergrenze und die globale Obergrenze so schnell wie möglich gegeneinander konvergieren zu lassen. Siehe auch Abbildung 4.4.

Inwiefern die globale Obergrenze die Größe des Baumes beeinflusst, wird bei Betrachtung der verschiedenen Gründe, wegen derer ein Knoten als untersucht gilt, ersichtlich.

Tritt einer der folgenden drei Fälle ein, gilt ein Teilproblem als vollständig untersucht und muss nicht weiter verzweigt werden:

- Unzulässigkeit:
  Durch das Hinzufügen der letzten Verzweigungsrestriktion ist das Teilproblem unzulässig geworden. Wenn es keine zulässige relaxierte Lösung für das Teilproblem gibt, gibt es auch keine zulässige ganzzahlige Lösung für das Teilproblem. Ein bereits als unzulässig erkanntes Teilproblem kann nachträglich nicht zulässig werden, da durch das Branchen auf Variablen der mögliche Lösungsraum zusätzlich verkleinert wird.

- Ganzzahlige Lösung:
  Wird innerhalb des Baumes eine ganzzahlige Lösung gefunden, gilt das Teilproblem als untersucht. Das heißt allerdings noch nicht, dass die ganzzahlige Lösung gleichzeitig die optimale Lösung darstellt. Die neu gefundene ganzzahlige Lösung wird lediglich mit der bisher besten z_{Best} verglichen und ggf. ausgetauscht.

- Überschreitung der Obergrenze:
  Ist das Ergebnis des relaxiert gelösten Teilproblems schlechter als die bisher beste IP-Lösung z_{Best}, müssen alle folgenden Teilprobleme nicht mehr betrachtet werden, da sich durch die zusätzlichen Einschränkungen der Zielfunktionswert nur noch verschlechtern kann.

Die optimale ganzzahlige Lösung ist erst dann gefunden, wenn wirklich alle Knoten abgearbeitet wurden.

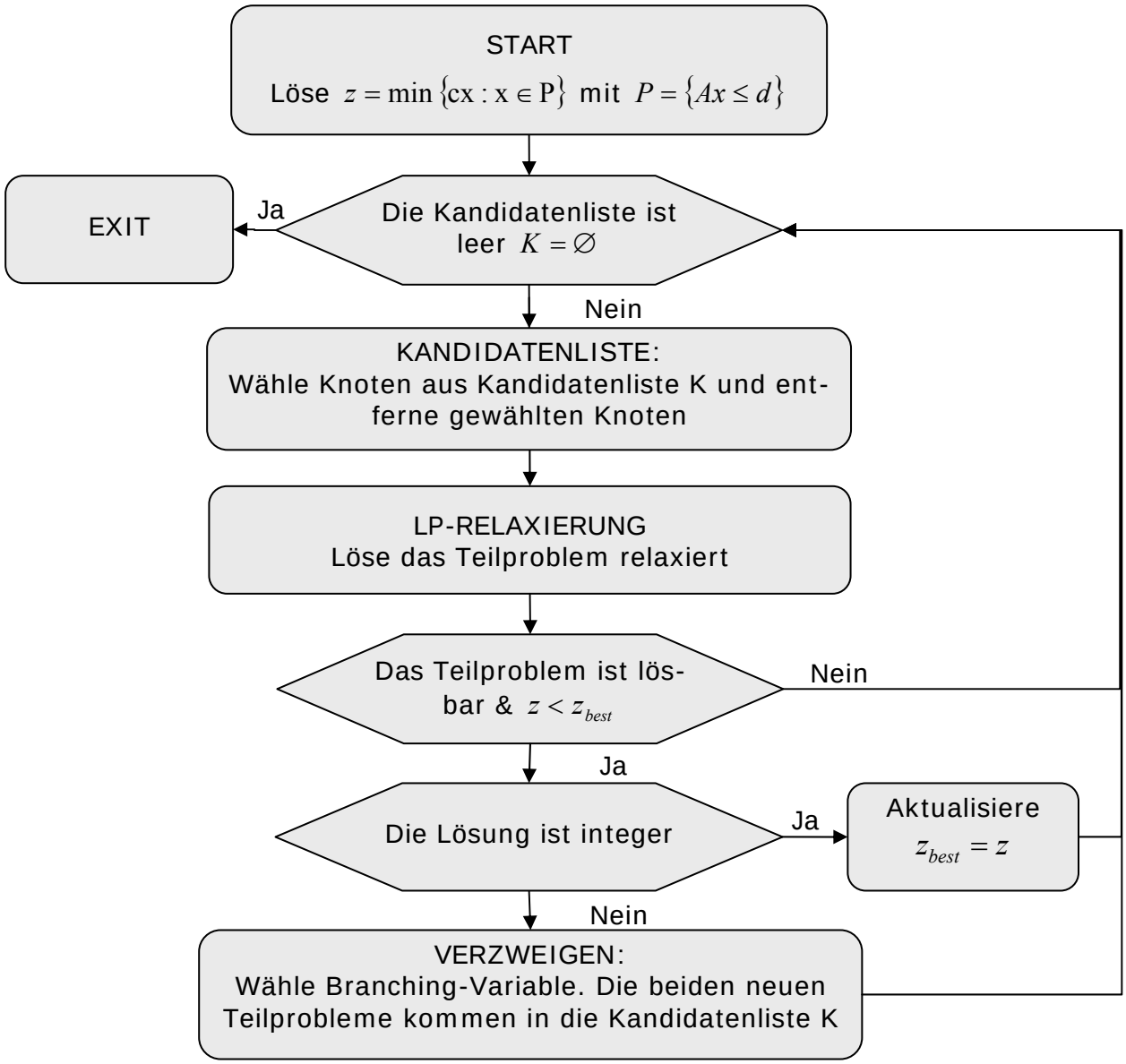

Abbildung 5.2: Das Branch-and-Bound Verfahren

Während der Bearbeitung der einzelnen Knoten, werden die noch nicht bearbeiteten Knoten in einer Kandidatenliste verwaltet. Nach bestimmten Auswahlregeln werden die in der Kandidatenliste hinterlegten Teilprobleme nacheinander bearbeitet. Für jeden Knoten wird eine Branching-Variable ausgewählt. Dadurch entstehen zwei neue Teilprobleme, die relaxiert gelöst werden. Wenn dann einer der drei oben beschriebenen Fälle erfüllt ist, gilt das entsprechende Teilproblem als vollständig untersucht und es wird nicht weiter verzweigt. Anderenfalls wird das Teilproblem der Kandidatenliste hinzugefügt. Erst wenn sich kein Teilproblem mehr in der Kandidatenliste befindet, ist die Optimalität einer ganzzahligen Lösung bewiesen.

Wurde bis zu diesem Zeitpunkt keine ganzzahlige Lösung gefunden, ist bewiesen, dass keine existiert. Abbildung 5.2 stellt eine Zusammenfassung des Verfahrens dar.

Bei der vorangegangenen Beschreibung des Branch-and-Bound Verfahrens bleiben zwei entscheidende Fragen offen:

- Nach welchen Regeln wird die Branching-Entscheidung getroffen?
- Nach welchen Regeln wird ein Knoten aus der Kandidatenliste ausgewählt?

Diese beiden Entscheidungen beeinflussen maßgeblich das Lösungsverhalten des Branch-and-Bound Prozesses und sollen aus diesem Grund in den folgenden Abschnitten besprochen werden.

5.2 Auswahlregeln für eine Branching-Variable

Aufgabe von Auswahlregeln ist die Ermittlung einer Branching-Variablen, über die ein Teilproblem verzweigt wird. Im Idealfall geht aus der Lösung der entsprechenden Teilprobleme eine neu globale Ober- oder Untergrenze hervor. Wie bereits beschrieben, verringert sich durch die gegenseitige Annäherung der Ober- und Untergrenze der Raum, in dem die optimale IP-Lösung liegen kann.

Im Folgenden soll ein kurzer Überblick über einige Auswahlregeln gegeben werden.

5.2.1 Verzweigen auf die fraktionellste Variable

Eine Variable, die in der relaxierten Lösung den fraktionellsten Wert annimmt, wird ausgewählt. Wenn x_j^i der Wert ist, den die Variable *j* an dem Knoten *i* hat, und $x_j^i = \lfloor x_j^i \rfloor + f_j^i$, wobei $f_j^i > 0$, dann wird eine Variable ausgewählt, die den größten Wert für MF_j^i hat, mit

$$MF_j^i = \min\left\{f_j^i, 1 - f_j^i\right\}.$$

Der Vorteil dieser Regel ist ihre Einfachheit. Leider gehen daraus keine guten Resultate hervor. In [AcKM04] werden sogar Ergebnisse präsentiert, die zeigen, dass eine zufällige Wahl der Branching-Variable vergleichbare Resultate erzielen kann.

5.2.2 Strong Branching

Hintergrund des von [ABCC95] vorgestellten *strong branchings* ist, dass die integer Variablen mit einem fraktionellen Wert an einem Knoten daraufhin getestet werden, welche von ihnen die größte Veränderung des Zielfunktionswertes hervorruft. Dazu wird auf die Variablen versuchsweise gebrancht, bevor eine Variable tatsächlich als Verzweigungs-Variable ausgewählt wird. Dieses Verfahren reduziert vor allem die Anzahl der Knoten. Leider kann jedoch die Zeit, die an einem Knoten verbraucht wird, aufgrund der vielen zu lösenden LP-Relaxierungen sehr lang sein. Um etwas Zeit zu sparen, kann die Menge der Variablen, auf die versuchsweise gebrancht wird, eingeschränkt werden oder es wird nicht die gesamte LP-Relaxierung durchgeführt. Nur eine bestimmte Anzahl an dualen Simplex Iterationen wird durchgeführt. Die Veränderung des Zielfunktionswertes lässt sich in der Regel schon nach wenigen Iterationen einschätzen.

5.2.3 Pseudocost Branching

Die Veränderung des Zielfunktionswertes, die mit dem Verzweigen einer bestimmten Variablen einhergeht, kann zumindest versuchsweise mittels der sogenannten Pseudocosts [BGGH71] prognostiziert werden. Unter Pseudocosts wird die Veränderung des Zielfunktionswertes im Verhältnis zur der Veränderung des Wertes einer integer Variablen bezeichnet. So können die Kosten prognostiziert werden, die entstehen, wenn eine Variable auf- oder abgerundet wird.

Wenn x_j^i der Wert ist, den die Variable *j* an dem Knoten *i* hat und $x_j^i = \lfloor x_j^i \rfloor + f_j^i$, mit $f_j^i > 0$, dann ergeben sich die Pseudocosts für diese Variable aus

$$PC_j^- = \frac{z^{i-} - z^i}{f_j^i} \text{ und } \quad PC_j^+ = \frac{z^{i+} - z^i}{1 - f_j^i}$$

Dabei steht z^{i-} für den Zielfunktionswert des Nachfolgerknotens von *i*, wenn der Wertebereich der Variable *j* nach oben beschränkt wird und z^{i+} dementsprechend, wenn der Wertebereich der Variablen *j* nach unten beschränkt wird.

So ergibt sich als Veränderung (wenn auf die Variable *j* gebrancht wird):

$$D_j^i = PC^- f_j^i \text{ und } U_j^i = PC^+ \left(1 - f_j^i\right)$$

Wie diese Pseudocosts initialisiert werden und ob sie immer wieder neu von Knoten zu Knoten berechnet werden müssen, wird in [LiSa99] detailliert dargestellt.

Für die hier vorgestellten Vorgehensweisen zur Auswahl einer Branching-Variable existieren sowohl verschiedene Abwandlungen als auch eine Reihe alternativer Vorgehensweisen. So wird zum Beispiel in einem von [PaCh03] präsentierten Verfahren versucht, nicht die Auswirkung des Verzweigens einer Variablen auf den Zielfunktionswert zu schätzen sondern die Auswirkung auf die aktiven Nebenbedingungen. Weitere Verfahren werden in [HoVi95] beschrieben.

5.3 Regeln zur Knotenauswahl

In der Kandidatenliste werden alle bisher noch nicht bearbeiteten Knoten gespeichert. Die Reihenfolge, in der die Knoten abgearbeitet werden, hat großen Einfluss auf die Laufzeit und auch auf den benötigten Speicherplatz. Bei der Auswahl eines Knotens sind zwei verschiedene Ziele abzuwägen. Zum einen ist es vorteilhaft, so schnell wie möglich eine gültige integer Lösung zu finden, was dazu führt, dass die globale Obergrenze herunter gesetzt werden kann. Zum andern soll auch die globale Untergrenze heraufgesetzt werden, damit die Optimalität einer gefundenen Lösung bewiesen werden kann. Es gibt verschiedene Knotenauswahlregeln, die eine Reihenfolge festlegen.

Sie lassen sich unterteilen in:

- statische Regeln
- auf Schätzungen basierende Regeln
- zwei-Phasen Regeln
- backtracking Regeln

5.3.1 Statische Regeln

Die Depth-First- Regel und die Best-First-Regel sind statische Knotenauswahlregeln. Bei der Depth-First-Regel, entwickelt sich der Enumerationsbaum in der Regel in die Tiefe. Dazu wird immer der Knoten ausgewählt, der gerade als letzter in die Kandidatenliste aufgenommen wurde (FIFO). Ein wesentlicher Vorteil dieser Knotenauswahlregel besteht darin, dass nur relativ wenige Knoten in der Kandidatenliste gespeichert werden müssen. Darüber hinaus kann das Teilproblem an einem Knoten sehr schnell berechnet werden, da es oft das gleiche Problem, wie das Vorgängerproblem, lediglich erweitert um eine Verzweigung, darstellt.

Der entscheidende Nachteil dieser Knotenauswahlstrategie ist, dass ggf. Äste des Enumerationsbaumes, welche nicht die optimale Lösung enthalten, bis in die Tiefe abgearbeitet werden. Wäre eine bessere globale Obergrenze bekannt, hätten diese Äste ggf. vorzeitig gelöscht werden können. Offensichtlich kann es so dazu kommen, dass sehr viele Teilprobleme gelöst werden müssen, was wiederum in langen Laufzeiten resultieren kann.

Bei der Best-First-Regel wird das Ziel verfolgt, die globale Untergrenze zu erhöhen. Ein Knoten mit der kleinsten unteren Grenze, d.h. ein Knoten, dessen LP- Relaxierung den kleinsten Zielfunktionswert hat, wird als Erster aus der Kandidatenliste ausgewählt. Bei Anwendung dieser Regel entwickelt sich der Baum eher in die Breite. Das hat wiederum zur Folge, dass die Probleme, die nacheinander gelöst werden, wenig Ähnlichkeit hinsichtlich der Verzweigungen miteinander haben. Dafür wird die Gefahr reduziert, dass lange in einem Teilbaum, welcher nicht die optimale Lösung enthält, gesucht wird.

5.3.2 Auf Schätzungen basierende Regeln

Weder bei der Best-first-Regel noch bei Deep-first-Regel werden die Knoten danach ausgewählt, ob sie möglicherweise zu einer gültigen ganzzahligen Lösung führen. Die *Best-Projection-Regel* [FoHT74] hat genau das zum Ziel. Mit $s^i = \sum_{j \in I} \min\left(f_j^i, 1 - f_j^i \right)$ wird die Summe der Unzulässigkeiten am Knoten i dargestellt. Das ist die Summe, die benötigt werden würde, wenn jede fraktionelle Integer Variable auf den nächsten ganzzahligen Wert auf- oder abgerundet werden sollte. Das Bewertungskriterium für die Best-Projection-Regel ergibt sich aus:

$$BP^i = z_{LP}^i + \left(\frac{z_{BEST} - z_{LP}^0}{s^0} \right) s^i$$

Wobei z_{LP}^i die relaxierte Lösung für das Teilproblem des Knotens i ist.

Ein Knoten, mit dem besten (kleinsten) BP^i wird ausgewählt. Dabei ist zu beachten, dass ein Wert für die globale Obergrenze z_{Best} bekannt sein muss.

Bei der Best-Projection-Regel wird allerdings nicht beachtet, wie der Einfluss der einzelnen Variablen auf s^i ist und wie hoch die entsprechenden Kosten für das Auf- oder Abrunden der Variablen sind. Mit Hilfe der Pseudokosten (s. Kapitel 5.2.3) kann diesem Aspekt Beachtung geschenkt werden. Durch die Einführung von Pseudokosten wird aus der Best-Projection-Regel die Best-Estimation-Regel.

$$BE^i = z_{LB}^i + \sum_{j \in I} \min\left(PC_j^- f_j, PC_j^+ (1 - f_j) \right)$$

Vorteil dieser Regel ist, dass kein Wert für die globale Obergrenze z_{Best} bekannt sein muss. Für diese Regel wird allerdings angenommen, dass sich eine gültige Integerlösung ergibt, wenn eine fraktionelle Integervariable auf den nächsten ganzzahligen Wert auf- oder abgerundet wird. Diese Annahme ist sicherlich nicht sehr realistisch. Eine realistischere Anpassung der Regel mittels Wahrscheinlichkeiten wird in [LiSa99] beschrieben.

5.3.3 Zwei-Phasen-Regeln

Wie bereits einleitend erwähnt, tritt bei der Auswahl eines Knotens ein Zielkonflikt auf. Er besteht darin, dass zum einen eine gute ganzzahlige Lösung gefunden werden soll und zum anderen bewiesen werden soll, dass keine bessere Lösung gefunden werden kann. Eine Kombination der bereits vorgestellten Regeln ist somit naheliegend. So kann die Depth-First-Regel angewandt werden, bis eine optimale Lösung gefunden ist. Danach kann zu der Best-First-Regel gewechselt werden, mittels der bewiesen werden soll, dass diese gefundene Lösung auch wirklich die beste ist.

Ein weiterer von [FoHT74] entwickelter Zwei-Phasen-Ansatz nutzt für die erste Phase die Best-Estimation-Regeln, bis eine gültige ganzzahlige Lösung gefunden wird. In der zweiten Phase soll ein Knoten gewählt werden, der den kleinsten *percentage error*:

$$PE^i = 100\frac{z_{Best} - BE^i}{z^i_{LP} - z_{Best}}.$$

aufweist. Dieser soll die Fehlerhaftigkeit einer Schätzung ausdrücken.

5.3.4 Backtracking Methoden

Es wird ein E_0 festgelegt, welches einen Schätzwert für den optimalen Zielfunktionswert z_{IP} darstellt. Dieser Wert gilt als Entscheidungskriterium. Ist $z^i < E_0$, soll die Depth-First-Regel zur Anwendung kommen. Sobald $z^i \geq E_0$, wird eine andere Regel benutzt. Damit soll das Abarbeiten von überflüssigen Knoten vermieden werden. Mit überflüssigen Knoten sind alle Knoten gemeint, die $z^i > z_{IP}$. Hauptkritikpunkt an der Depth-First-Regel war, dass ggf. sehr viele überflüssige Knoten abgearbeitet werden. Dies kann durch eine Backtracking Methode vermieden bzw. verringert werden. Es sollen somit nur die Vorteile der Depth-First-Regel bezüglich des Speicherplatzes und der schnellen Lösung der einzelnen Teilprobleme zum Tragen kommen. Die Backtracking Methoden unterscheiden sich sowohl hinsichtlich der Festlegung von E_0 ,wofür eine der auf Schätzungen basierenden Methoden herangezogen werden kann, als auch hinsichtlich der zweiten Regel, die zur Anwendung kommen soll.

Vergleiche der einzelnen Knotenauswahlregeln sind in [LiSa99] zu finden. Leider kann keine Knotenauswahlregel allgemeingültig als die beste identifiziert werden. Eine Knotenauswahlregel, die für ein Problem besonders vorteilhaft ist, kann für ein anderes besonders unvorteilhaft sein. Daher sollte ein Optimierer immer die Möglichkeit gewähren, zwischen verschiedenen Knotenauswahlregeln zu wählen. Insgesamt scheinen allerdings die Regeln, die mit Pseudocosts arbeiten, oft eine gute Wahl zu sein.