3 Ganzzahlige Optimierung

In vielen Anwendungen sind die Variablen einer linearen Optimierungsaufgabe der zusätzlichen Nebenbedingung unterworfen, dass sie ganzzahlig sein müssen. Dies ist etwa der Fall, wenn bei einer Transportaufgabe die Güter (z. B. Autos, Kontainer, etc.) nur in festen Einheiten auftreten und nicht beliebig teilbar sind. Im Extremfall, etwa bei Entscheidungsprozessen, können auch nur die Variablenwerte 0 und 1 auftreten. Hier und bei Lösungen mit kleinen Stückzahlen verbietet es sich von selbst, eine *ganzzahlige* Lösung einfach durch Rundung einer „*kontinuierlichen*" erzeugen zu wollen. Die Theorie und Numerik der ganzzahligen Optimierung sind wesentlich komplizierter und aufwendiger als die der gewöhnlichen. Es existiert eine Vielzahl von speziellen Resultaten und Methoden, die auf bestimmte Problemklassen zugeschnitten sind. Wir betrachten im Folgenden nur das Transportproblem als ein Beispiel der ganzzahligen Programmierung ohne Komplikationen. Für allgemeinere Aufgaben wird der Algorithmus von Gomory[1] (1958); beschrieben, der eine „ganzzahlige" Erweiterung des uns schon bekannten Simplex-Verfahrens darstellt.

3.1 Das klassische Transportproblem

Gegeben sei eine Händlerorganisation mit m Warenlagern und n Verbrauchsorten. Gefragt ist nach einem kostenminimalen Verteilungsmodus. Wir verwenden die folgenden Bezeichnungen:

- Lager: F_i, Bestand Einheiten/Monat a_i $(i = 1, \dots, m)$;
- Orte: G_i, Bedarf Einheiten/Monat b_j $(j = 1, \dots, n)$;
- Transportierte Menge $F_i \to G_j$: x_{ij} Einheiten (ganzzahlig!);
- Transportkosten pro Einheit $F_i \to G_j$: c_{ij} EURO.

Bedingung der Bedarfsdeckung:

$$\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j. \tag{3.1.1}$$

[1]Ralph E. Gomory (1929–): US-amerikanischer Informatiker und angewandter Mathematiker; arbeitete zunächst an der Universität Princeton und ab 1959 in leitender Position bei IBM, ab 1989 Präsident der Alfred P. Sloan Foundation; Beiträge zur linearen Optimierung, Netzwerktheorie und nichtlineare Differentialgleichungen; hauptsächlich bekannt durch die nach ihm benannte Methode des „Gomory-Schnitts" in der ganzzahligen Optimierung: "Outline of an algorithm for integer solutions to linear programs", Bulletin of the American Mathematical Society 64, 275–278 (1958).

Formulierung als lineare Programmierungsaufgabe:

$$x = (x_{ij})_{i,j=1}^{m,n}: \qquad \sum_{i,j=1}^{m,n} c_{ij} \to \min!$$

$$x_{ij} \geq 0 \quad \text{ganzzahlig} \ \ (i = 1,\dots,m; j = 1,\dots,n), \tag{3.1.2}$$

$$\sum_{j=1}^{n} x_{ij} = a_i \ \ (i = 1,\dots,m), \quad \sum_{i=1}^{m} x_{ij} = b_j \ \ (j = 1,\dots,n).$$

Ausgeschrieben lauten die Nebenbedingungen wie folgt:

$$\begin{bmatrix} x_{11}+\ldots+x_{1n} & & & \\ & x_{21}+\ldots+x_{2n} & & \\ & & \ddots & \\ & & & x_{m1}+\ldots+x_{mn} \\ x_{11}+ & x_{21}+ & \ldots & x_{m1} \\ \quad x_{12}+ & \quad x_{22}+ & \ldots & \quad x_{m2} \\ \quad\ddots & \quad\ddots & & \quad\ddots \\ \qquad x_{1n}+ & \qquad x_{2n}+ & \ldots & \qquad x_{mn} \end{bmatrix} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \\ b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}.$$

Dieses Problem hat offensichtlich einen beschränkten (und damit kompakten) zulässigen Bereich und ist folglich „kontinuierlich“ lösbar. Wir wollen zeigen, dass die Lösung auch ganzzahlig ist, wenn es die a_i und b_j sind. Wegen der Verträglichkeitsbedingung (3.1.1) sind höchstens $m+n-1$ der Nebenbedingungen linear unabhängig. Eine einfache Rangbetrachtung zeigt, dass es genau $m+n-1$ linear unabhängige Spaltenvektoren in der zugehörigen Koeffizientenmatrix gibt. Folglich kann irgendeine der $m+n$ Gleichungen als redundant weggelassen werden; im Folgenden wird es o.B.d.A. die letzte sein. Das verbleibende System hat dann den Höchstrang $m+n-1$ und die Programmierungsaufgabe hat $(a_i \geq 0,\ b_j \geq 0)$ kanonische Form.

Lemma 3.1: *Jede Basis von Spaltenvektoren des um die letzte Zeile reduzierten Systems hat nach geeigneter Umordnung von Spalten und Zeilen Dreiecksgestalt.*

Beweis: Offenbar muss bei einer Basis $[a_{k_1},\dots,a_{k_{m+n-1}}]$ von Spaltenvektoren in jeder Zeile mindestens eine 1 stehen. Andererseits muss es unter den ersten m oder unter den letzten $n-1$ Zeilen mindestens eine geben, in der *genau eine* 1 steht. Andernfalls müsste die Anzahl der Spalten $k \geq 2m$ bzw. $k \geq 2(n-1)$ und somit $2k \geq 2m+2(n-1)+1$ sein, da eine der Spalten mit von Null verschiedenem Eintrag in der gestrichenen letzten Zeile auftreten muss. Dies steht aber im Widerspruch zum Rang $m+n-1$. Wir streichen diese eine Zeile sowie die zugehörige Spalte aus dem System; das reduzierte System hat dann den Rang $m+n-2$. Sukzessive Anwendung des obigen Arguments auf die entstehenden reduzierten Systeme ergibt schließlich die Behauptung. Q.E.D.

Satz 3.1: *Sind beim Transportproblem die* a_i $(i = 1, \ldots, m)$ *und* b_j $(j = 1, \ldots, n)$ *ganzzahlig, so ist auch jede vom Simplex-Verfahren gelieferte Basislösung ganzzahlig.*

Beweis: Jede Basislösung ist gegeben durch ein lineares $(m + n - 1) \times (m + n - 1)$-Gleichungssystem mit der aus den zugehörigen Spaltenvektoren der Ausgangsmatrix gebildeten Koeffizientenmatrix und dem Vektor $(a_1, \ldots, a_m, b_1, \ldots, b_{n-1})^T$ als rechte Seite. Nach Lemma 3.1 hat dieses System Dreicksgestalt und kann folglich sukzessive aufgelöst werden. Dabei treten aber stets nur Divisionen durch 1 auf, so dass die Lösung zwangsläufig ganzzahlig ist. Q.E.D.

3.2 Das duale Simplex-Verfahren

Als Vorbereitung für das Verfahren von Gomory zur Lösung ganzzahliger Programmierungsaufgaben beschreiben wir im Folgenden eine Variante des „primalen“ Simplex-Verfahrens, den sog. „dualen“ Simplex-Algorithmus. Ausgangspunkt ist die folgende duale Standardaufgabe:

$$(I^*) \qquad c^T \cdot x \to \min!, \quad Ax \geq b, \quad x \geq 0.$$

Es sei $c \geq 0$, aber für b werden keine Vorzeichenbedingungen gestellt. üblicherweise würde man zur Lösung dieser Aufgabe zunächst zu ihrer dualen übergehen:

$$(I) \qquad b^T \cdot y \to \max!, \quad A^T y \leq c, \quad y \geq 0,$$

und diese dann nach Einführung von Schlupfvariablen $v \geq 0$ mit dem üblichen Simplex-Verfahren behandeln. Aus dem optimalen Tableau ließe sich am Schluss eine Lösung von (I^*) nach der in Abschnitt 2.2.1 beschriebenen Methode gewinnen. Alternativ lässt sich Problem (I^*) aber auch in der folgenden Form schreiben:

$$(\tilde{I}^*) \qquad c^T \cdot x \to \min!, \quad -Ax \leq -b, \quad x \geq 0,$$

die nach Einführung von Schlupfvariablen $u \geq 0$ übergeht in

$$(\tilde{I}^*) \qquad c^T \cdot x \to \min!, \quad -Ax + Iu = -b, \quad x \geq 0,\ u \geq 0.$$

Das Problem bei der Lösung dieser Aufgabe mit dem Simplex-Verfahren besteht darin, dass im Falle $-b \not\geq 0$ durch $(x^0, u^0)^T = (0, -b)^T$ nicht automatisch eine zulässige Startecke gegeben ist. An dieser Stelle setzt der „duale“ Simplex-Algorithmus an; er erzeugt im Gegensatz zum „primalen“ Simplex-Algorithmus eine endliche Folge von Punkten, die zwar für (I^*) nicht zulässig sind, für deren Zielfunktionalwert aber stets

$$c^T \cdot x^0 \leq c^T \cdot x, \quad x \in M^*,$$

gilt. Nach endlich vielen Austauschschritten wird dann ein zulässiger Eckpunkt $\hat{x} \in M^*$ erreicht, der zwangsläufig optimal für (I^*) ist.

Ausgangspunkt: Wir betrachten das kanonische Programmierungsproblem

$$(\tilde{I}) \qquad c^T \cdot x \to \min!, \quad Ax = b, \;\; x \geq 0,$$

mit $c \geq 0$ (aber nicht notwendig $b \geq 0$). Sei $I^0 \subset \{1, \dots, n\}$ und $A_0 = [a_i, \, i \in I^0]$ regulär. Der durch $x^0 = A_0^{-1}b$ und $x_i^0 := 0, \, i \notin I^0$ bestimmte Vektor des $\mathbb{R}^n$ löst zwar die Gleichung $Ax^0 = b$, doch ist er i. Allg. wegen $x^0 \ngeq 0$ nicht zulässig. Analog zur Herleitung des normalen Simplex-Verfahrens erhalten wir die Gleichungen

$$\begin{aligned} x_i &= \sum_{k \notin I^0} \alpha_{ik} x_k + x_i^0 \quad (i \in I^0), \\ z &= \sum_{k \notin I^0} \gamma_k x_k + c^T \cdot x^0, \quad \gamma_k = \sum_{i \in I^0} \alpha_{ik} c_i + c_k, \quad k \notin I^0, \end{aligned}$$

die notwendig (und hinreichend) von jedem $x \in \mathbb{R}^n$ mit $Ax = b$ erfüllt werden. Das zugehörige Simplex-Tableau ist:

	$x_k \; (k \notin I^0)$	
x_i $(i \in I^0)$	α_{ik} $(i \in I^0, \; k \notin I^0)$	x_i^0 $(i \in I^0)$
z	$\gamma_k \; (k \notin I^0)$	$c^T \cdot x^0$

Im Normalfall $x_i^0 \geq 0$ ist x^0 zulässige Ecke und für $\gamma_k \geq 0$ notwendig optimal. Wir nehmen jetzt an, dass am Anfang $\gamma_k \geq 0$ ist (aber nicht notwendig $x_i^0 \geq 0$), d. h.:

$$z = c^T \cdot x \geq c^T \cdot x^0, \quad x \in M^*.$$

Der „duale “ Simplex-Algorithmus sucht nun durch Basiswechsel die negativen Komponeneten von x^0 zu beseitigen, ohne die Eigenschaft $\gamma_k \geq 0$ zu verlieren.

Der Algorithmus:

1. Gilt $x_i^0 \geq 0 \; (i \in I^0)$, so ist x^0 zulässig und wegen

$$c^T \cdot x = \sum_{k \notin I^0} \gamma_k x_k + c^T \cdot x^0 \geq c^T \cdot x^0, \quad x \in M^*,$$

auch optimal.

2. Gibt es ein $p \in I^0$ mit $x_p^0 < 0$, so sind zwei Fälle zu unterscheiden:

a) Im Falle $\alpha_{pk} \leq 0 \; (k \notin I^0)$ folgt für $x \in M$ der Widerspruch

$$0 \leq x_p = \sum_{k \notin I^0} \alpha_{pk} x_k + x_p^0 < 0,$$

d. h.: $M = \emptyset$ und die Aufgabe ist folglich nicht lösbar.

b) Gibt es Indizes $q \notin I^0$ mit $\alpha_{pq} > 0$, so wird die Variable x_p gegen eine noch auszuwählende der x_q ausgetauscht. Man erhält so wieder eine Basis. Das Auswahlkriterion ist:

$$(R^*) \qquad q \in I^0 : \quad \frac{\gamma_q}{\alpha_{pq}} = \min_{\alpha_{pk}>0} \frac{\gamma_k}{\alpha_{pk}} \geq 0.$$

Der Austauschschritt, d. h. der Übergang von der Indexmenge I^0 zur Indexmenge $\tilde{I}^0 = (I^0 \setminus \{p\}) \cup \{q\}$ mit entsprechendem Basiswechsel, verläuft wie folgt:

- Pivotelement:

$$\alpha_{pq} \;\to\; \tilde{\alpha}_{qp} = \frac{1}{\alpha_{pq}};$$

- Pivotspalte:

$$\begin{aligned} \alpha_{iq} \;&\to\; \tilde{\alpha}_{ip} = \frac{\alpha_{iq}}{\alpha_{pq}} \quad (i \in I^0,\ i \neq p), \\ \gamma_q \;&\to\; \tilde{\gamma}_p = \frac{\gamma_q}{\alpha_{pq}}; \end{aligned}$$

- Pivotzeile:

$$\begin{aligned} \alpha_{pk} \;&\to\; \tilde{\alpha}_{qk} = -\frac{\alpha_{pk}}{\alpha_{pq}} \quad (k \notin I^0,\ k \neq q), \\ x_p^0 \;&\to\; \tilde{x}_p^0 = -\frac{x_p^0}{\alpha_{pq}}; \end{aligned}$$

- Sonstige:

$$\begin{aligned} \alpha_{ik} \;&\to\; \tilde{\alpha}_{ik} = \alpha_{ik} - \frac{\alpha_{iq}\alpha_{pk}}{\alpha_{pq}} \quad (i \in I^0, i \neq p,\ k \notin I^0, k \neq q), \\ \gamma_k \;&\to\; \tilde{\gamma}_k = \gamma_k - \frac{\gamma_q\alpha_{pk}}{\alpha_{pq}} \quad (k \notin I^0), \\ x_i^0 \;&\to\; \tilde{x}_i^0 = x_i^0 - \frac{\alpha_{iq}x_p^0}{\alpha_{pq}} \quad (i \in I^0), \\ c^T \cdot x^0 \;&\to\; c^T \cdot \tilde{x}^0 = c^T \cdot x^0 - \frac{\gamma_q x_p^0}{\alpha_{pq}}. \end{aligned}$$

Satz 3.2: *Ist zu Beginn $\gamma_k \geq 0$ $(k \notin I^0)$, so liefert der „duale“ Simplex-Algorithmus mit der Auswahlregel (R^*) in endlich vielen Austauschschritten eine Lösung der Programmierungsaufgabe $(\tilde{I})$ oder erweist ihre Unlösbarkeit, wenn keine Zyklen auftreten.*

Beweis: Im Falle $\gamma_k \geq 0$ $(k \notin I^0)$ ist nach dem Austauschschritt:

$$\tilde{\gamma}_p = \frac{\gamma_q}{\alpha_{pq}} \geq 0, \quad \tilde{x}_q^0 = -\frac{x_p^0}{\alpha_{pq}} > 0,$$

und

$$\begin{aligned} &a) \qquad \alpha_{pk} \leq 0 : \quad \tilde{\gamma}_k = \gamma_k + \frac{\gamma_q}{\alpha_{pq}}(-\alpha_{pk}) \geq \gamma_k \geq 0, \\ &b) \qquad \alpha_{pk} > 0 : \quad \tilde{\gamma}_k = \alpha_{pk}\Big(\frac{\gamma_k}{\alpha_{pk}} - \frac{\gamma_q}{\alpha_{pq}}\Big) \geq 0, \end{aligned}$$

sowie

$$c^T \cdot \tilde{x}^0 = c^T \cdot x^0 + \frac{\gamma_q}{\alpha_{pq}}(-x_p^0) \begin{cases} > c^T \cdot x^0 & (\gamma_q > 0) \\ = c^T \cdot x^0 & (\gamma_q = 0) \end{cases}.$$

Jeder Austauschschritt bewirkt also eine Vergrößerung (bzw. Nicht-Verkleinerung) des Zielfunktionalwerts unter Beibehaltung der Eigenschaft $\gamma_k \geq 0$ $(k \in I^0)$. Wenn Zyklen ausgeschlossen sind, d. h. wenn jede Basis höchstens einmal auftritt, muss also in endlich vielen Schritten ein Tableau erreicht werden, das zu einer Basis gehört, für die der zugehörige Punkt $x^0 \in M$ ist, wenn ein solcher überhaupt existiert. Wegen

$$c^T \cdot x \geq c^T \cdot x^0, \quad x \in M,$$

ist dieser Punkt dann notwendig Lösung der Aufgabe $(\tilde{I})$. Q.E.D.

Zum sicheren Ausschluss von Zyklen kann die Auswahlregel (R^*) geeignet verfeinert werden zu $(\tilde{R}^*)$ (analog zum normalen Simplex-Algorithmus).

3.3 Das Schnittverfahren von Gomory

Wir betrachten nun das übliche kanonische Problem

$$(II) \qquad c^T \cdot x \to \min!, \quad Ax = b, \;\; x \geq 0,$$

mit $b \geq 0$ und der Zusatzbedingung $x \in \mathbb{Z}^n$. Man löse zunächst Aufgabe (II) ohne Berücksichtigung der Bedingung $x \in \mathbb{Z}^n$ mit Hilfe des Simplex-Verfahrens. I. Allg. wird die gefundene Lösung $\hat{x}$ nicht ganzzahlig sein. Der zulässige Bereich M wird durch eine zusätzliche Bedingung (einen „Schnitt") so eingeschränkt, dass $\hat{x}$ nicht mehr enthalten ist, aber alle vorher zulässigen *ganzzahligen* Punkte zulässig bleiben. Geometrisch bedeutet das, dass eine Hyperebene bestimmt wird, welche die nichtganzzahlige, optimale Ecke $\hat{x}$ von den übrigen zulässigen, ganzzahligen Punkten trennt (s. Abb. 3.3).

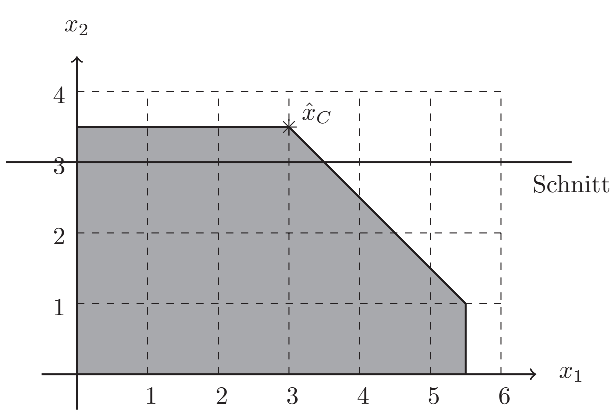

Abbildung 3.1: *Schema der Gomoryschen Schnitt-Methode*

Das Problem wird dann mit der neuen Menge M wieder mit dem Simplex-Verfahren gelöst, u.s.w.. Durch die Schnitte wird die zulässige Menge nach und nach auf die konvexe Hülle ihrer ganzzahlichen Punkte reduziert. Man kann zeigen, dass es möglich ist, so in endlich vielen Schritten ein ganzzahliges Minimum (bezogen auf alle ganzzahligen Punkte) zu finden, wenn ein solches überhaupt existiert.

Die Anwendung des normalen Simplex-Algorithmus auf Problem (II) (Phase I + II) führe auf das Endtableau

$$\begin{array}{c|c|c}
 & x_k & \\
\hline
x_i & \alpha_{ik} & x_i^0 \\
\hline
 & \gamma_k & c^T \cdot x^0
\end{array}$$

mit $\gamma_k \geq 0$ $(k \notin I^0)$ und $x_i^0 \geq 0$ $(i \in I^0)$. Dieses beinhaltet die zur Ausgangsgleichung $Ax = b$ äquivalenten Gleichungen

$$x_i = \sum_{k \notin I^0} \alpha_{ik} x_k + x_i^0 \quad (i \in I^0), \tag{3.3.3}$$

und die Beziehung

$$c^T \cdot x = \sum_{k \notin I^0} \gamma_k x_k + c^T \cdot x^0, \quad x \in M. \tag{3.3.4}$$

Im Falle $x_i^0 \in \mathbb{Z}$ $(i \in I^0)$ ist x^0 ganzzahlige Lösung und folglich auch Lösung der ganzzahligen Programmierungsaufgabe.

Für Zahlen $a \in \mathbb{R}$ wird die Zerlegung $a = [a] + \{a\}$ in ihren „ganzzahligen“ Anteil $[a] \in \mathbb{Z}$ und ihren „gebrochenen“ Anteil $\{a\} \in [0,1)$ eingeführt. Sei nun $x_p^0 \notin \mathbb{Z}$ für ein $p \in I^0$. Für alle Punkte $x \in M \cap \mathbb{Z}^n$ gilt gemäß (3.3.3):

$$x_p + \sum_{k \notin I^0} [-\alpha_{pk}] x_k - [x_p^0] = -\sum_{k \notin I^0} \{-\alpha_{pk}\} x_k + \{x_p^0\}. \tag{3.3.5}$$

Die linke Seite ist offenbar ganzzahlig und damit auch die rechte Seite. Ferner gilt

$$0 \leq \{-\alpha_{pk}\},\ \{x_p^0\} < 1, \quad -\sum_{k \notin I^0} \{-\alpha_{pk}\} x_k \leq 0.$$

Dies impliziert notwendig, dass

$$-\sum_{k \notin I^0} \{-\alpha_{pk}\} x_k + \{x_p^0\} < 1$$

und folglich, wegen der Ganzzahligkeit des Terms,

$$-\sum_{k \notin I^0} \{-\alpha_{pk}\} x_k + \{x_p^0\} \leq 0.$$

Für Punkte $x \in M \cap \mathbb{Z}^n$ muss also gelten:

$$(S) \qquad -\sum_{k \notin I^0} \{-\alpha_{pk}\} x_k + \{x_p^0\} \leq 0. \tag{3.3.6}$$

Für den optimalen Punkt x^0 mit $x_p^0 \notin \mathbb{Z}$ gilt dagegen wegen $x_i^0 = 0,\ i \notin I^0$:

$$-\sum_{k \notin I^0} \{-\alpha_{pk}\} x_k + \{x_p^0\} = \{x_p^0\} > 0. \tag{3.3.7}$$

Die Zusatzbedingung (S) „trennt“ also den Punkt x^0 von allen zulässigen, ganzzahligen Punkten. Damit haben wir das folgende Lemma bewiesen.

Lemma 3.2: *Jeder Punkt $x \in M \cap \mathbb{Z}^n$ erfüllt die Schnittbedingung (S), während der optimale, aber nichtganzzahlige Punkt $x^0 \in M$ sie nicht erfüllt.*

Die Schnittbedingung (S) wird nun durch Einführung einer Schlupfvariablen $s_1 \geq 0$ in eine Gleichungsnebenbedingung umgeformt,

$$s_1 - \sum_{k \notin I^0} \{-\alpha_{pk}\} x_k = -\{x_p^0\}, \tag{3.3.8}$$

und diese dem obigen Endtableau hinzugefügt. Die neue Variable s_1 wird also „Basisvariable“, allerdings mit dem nicht zulässigen Wert $s_1 = -\{x_p^0\} < 0$. Das so gewonnene Tableau ist *kein* zulässiges Ausgangstableau für die normale Simplex-Methode, wohl aber für ihr *duales* Gegenstück (Man beachte, dass $\gamma_k \geq 0, k \notin I^0$.). Dies ist die eigentliche Motivation für die Einführung der *dualen* Simplex-Methode.

Ausgehend vom erweiterten Tableau

	x_k	
x_i	α_{ik}	x_i^0
s_1	$\{-\alpha_{pk}\}$	$-\{x_p^0\}$
	γ_k	$-c^T \cdot x^0$

werden nach den Regeln der dualen Simplex-Methode Basiswechsel vorgenommen, bis alle Einträge in der letzten Spalte wieder nicht negativ sind. Dabei bleibt $\gamma_k \geq 0$, und der Zielfunktionalwerte wird nicht verkleinert. Man erhält also ein „optimales“ Tableau für die durch einen Gomory-Schnitt modifizierte (nicht ganzzahlige) Aufgabe (II). Die ausführliche Konvergenzanalyse dieses Algorithmus kann hier nicht gegeben werden, doch soll die Vorgehensweise an einem Beispiel illustriert werden. Bei der Wahl des Index $p \in I^0$ für den Schnitt wird im Folgenden nach der Regel vorgegangen, den mit größtem gebrochenen Anteil $\{x_p^0\}$ zu nehmen.

Beispiel 3.1: Wir betrachten die folgende ganzzahlige Programmierungsaufgabe:

$$(\tilde{I}) \qquad x \in \mathbb{Z}_+^2 : \quad \begin{aligned} 3x_1 + x_2 &\to \max! \\ x_1 + 2x_2 &\leq 8, \\ 3x_1 - 4x_2 &\leq 12. \end{aligned}$$

Nach Einführung von Schlupfvariablen x_3, x_4 geht diese Aufgabe über in die kanonische Form:

$$(\tilde{II}) \qquad x \in \mathbb{Z}_+^4 : \quad \begin{aligned} -3x_1 - x_2 &\to \min! \\ x_1 + 2x_2 + x_3 &= 8, \\ 3x_1 - 4x_2 + x_4 &= 12. \end{aligned}$$

i) Normales Simplex-Verfahren: Anwendung des normalen Simplex-Algorithmus mit Auswahlregel (R) ergibt die folgenden Tableaus:

	x_1	x_2		γ_q/α_{iq}
x_3	-1	-2	8	-8
x_4	$\boxed{-3}$	4	12	-4
	$\boxed{-3}$	-1	0	

$\to$

	x_4	x_2	
x_3	$1/3$	$\boxed{-10/3}$	4
x_1	$-1/3$	$4/3$	4
	1	$\boxed{-5}$	-12

$\to$

	x_4	x_3	
x_2	$1/10$	$-3/10$	$6/5$
x_1	$-1/5$	$-2/5$	$28/5$
	$1/2$	$3/2$	-18

Wir entnehmen die „kontinuierliche" Lösung $(x_1, x_2)^T = (28/5, 6/5)^T$ mit dem zugehörigen Zielfunktionalwert $z = 18$.

a) Rundung der „kontinuierlichen" Lösung auf Ganzzahligkeit ergibt:

$$(\tilde{x}_1, \tilde{x}_2)^T = (6, 1)^T, \quad z = 19 \ \text{(nicht zulässig)};$$

b) Abschneiden der „kontinuierlichen" Lösung auf Ganzzahligkeit ergibt:

$$(\bar{x}_1, \bar{x}_2)^T = (5, 1)^T, \quad z = 16 \ \text{(zulässig)};$$

ii) Die Gomorysche Schnitt-Methode führt auf den ersten Schnitt:

$$s_1 - \frac{1}{5}x_4 - \frac{2}{5}x_3 = -\frac{3}{5}$$

	x_4	x_3	
x_2	$1/10$	$-3/10$	$6/5$
x_1	$-1/5$	$-2/5$	$28/5$
s_1	$\boxed{1/5}$	$2/5$	$\boxed{-3/5}$
	$1/2$	$3/2$	-18
γ_k/α_{ik}	$5/2$	$15/4$	

$\to$

	s_1	x_3	
x_2	$1/2$	$-1/2$	$3/2$
x_1	-1	0	5
x_4	5	-2	3
	$5/2$	$1/2$	$-33/2$

Als zweiter Schnitt ergibt sich $(p = 2)$:

$$s_2 + \frac{1}{2}s_1 - \frac{1}{2}x_3 = -\frac{1}{2}$$

$$\begin{array}{c|cc|c}
 & s_1 & x_3 & \\
\hline
x_2 & 1/2 & -1/2 & 3/2 \\
x_1 & -1 & 0 & 5 \\
x_4 & 5 & -2 & 3 \\
s_2 & 1/2 & 1/2 & -1/2 \\
\hline
 & 5/2 & 1/2 & -33/2 \\
\hline
\gamma_k/\alpha_{ik} & 5 & 1 &
\end{array}
\quad \rightarrow \quad
\begin{array}{c|cc|c}
 & s_1 & s_2 & \\
\hline
x_2 & * & -1 & 1 \\
x_1 & * & 0 & 5 \\
x_4 & * & -4 & 1 \\
x_3 & -1 & 2 & 1 \\
\hline
 & 2 & 1 & -16 \\
\hline
 & & &
\end{array}$$

Die gewonnene ganzzahlige Lösung ist $(x_1, x_2)^T = (5, 1)^T$ mit dem Wert $z = 16$.

3.4 Übungsaufgaben

Aufgabe 3.1: Für die Programmierungsaufgabe

$$c^T \cdot x \to \min!, \quad Ax = b, \;\; x \geq 0,$$

sei ein Simplextableau zur Basis $B = \{a_i,\ i \in I^0\}$ gegeben. Die Einträge x_i^0 $(i \in I^0)$ der letzten Spalte und die γ_k $(k \neq I^0)$ in der letzten Zeile brauchen dabei nicht notwendig nicht-negativ zu sein.

Man zeige, dass für jedes Pivotelement $\alpha_{pq} \neq 0,\ p \in I^0, q \neq I^0$, automatisch $\tilde{B} = \{a_i,\ i \in \tilde{I}^0 = (I^0 \setminus \{p\}) \cup \{q\}\}$ wieder eine Basis ist. Der Austauschschritt des *primalen* (für $\alpha_{pq} < 0$) bzw. des *dualen* (für $\alpha_{pq} > 0$) Simplex-Algorithmus führt dann auf ein Tableau zu dieser neuen Basis $\tilde{B}$.

Aufgabe 3.2: Man löse die Programmierungsaufgabe

$$\begin{aligned}
5x_1 + 6x_2 &\to \min! \\
x_1 \geq 0,\ x_2 &\geq 0, \\
2x_1 + x_2 &\geq 6, \\
2x_1 + 4x_2 &\geq 12,
\end{aligned}$$

mit Hilfe des *dualen* Simplex-Algorithmus.

Aufgabe 3.3: Das klassische Transportproblem aus dem Text beinhaltet die Gleichungsnebenbedingung

$$Ax = (a, b)^T,$$

für dem Zustandsvektor $x_{i,j=1}^{m,n} = (x_{11}, \dots, x_{m1}, \dots, x_{1n}, \dots, x_{mn})^T$ der transportierten Mengen mit den Vektoren $a = (a_i)_{i=1}^m$, $b = (b_j)_{j=1}^n$ und der folgenden Matrix $A \in \mathbb{R}^{m \times n}$:

$$A = \begin{bmatrix}
x_{11} & \dots & x_{1n} & & & \\
& x_{21} & \dots & x_{2n} & & \\
& & \ddots & & & \\
& & & x_{m1} & \dots & x_{mn} \\
x_{11}\,+ & x_{21}\,+ & \dots & x_{m1} & & \\
x_{12}\,+ & x_{22}\,+ & \dots & x_{m2} & & \\
\ddots & \ddots & & \ddots & & \\
x_{1n}\,+ & x_{2n}\,+ & \dots & x_{mn} & &
\end{bmatrix}$$

Man verifiziere, dass diese Matrix den $\mathrm{Rang}(A) = m + n - 1$ hat.

Aufgabe 3.4: Man löse die ganzzahlige Programmierungsaufgabe

$$\begin{aligned}
x \in \mathbb{Z}_+^2 : \quad 3x_1 - x_2 &\to \max! \\
3x_1 - 2x_2 &\le 3, \\
-5x_1 - 4x_2 &\le -10, \\
2x_1 + x_2 &\le 5,
\end{aligned}$$

mit Hilfe der Gomoryschen Schnitt-Methode und vergleiche das Ergebnis mit dem durch Abschneiden der „kontinuierlichen“ Lösung gewonnenen.