8 Lineare Optimierung

8.1 Lineare Programme

„Lineares Programm“ ist die historisch begründete Bezeichnung für lineare Optimierungsaufgaben vom folgenden Typ:

Beispiel 8.1: Eine Fabrik kann zwei Typen A und B eines Produkts unter folgenden Bedingungen herstellen:

Produkt	Typ A	Typ B	maximal möglich
Stück pro Tag	x_1	x_2	100 Stück
Arbeitszeit pro Stück	4	1	160 Stunden
Kosten pro Stück	20	10	1100 DM
Gewinn pro Stück	120	40	? DM

Wie müssen x_1 und x_2 gewählt werden, damit der Gewinn maximal wird? Dabei muss offenbar der lineare Ausdruck

$$Q(x_1, x_2) := 120x_1 + 40x_2$$

zu einem Maximum gemacht werden unter den linearen Nebenbedingungen

$$\begin{aligned} x_1 + \quad x_2 &\leq \ 100 \\ 4x_1 + \quad x_2 &\leq \ 160 \quad , \quad x_1 \geq 0\,, \quad x_2 \geq 0. \\ 20x_1 + 10x_2 &\leq 1100 \end{aligned} \tag{8.1.1}$$

Dies ist ein lineares Programm in sog. „Standardform“.

Beispiel 8.2: Die Produktion von 7 Zuckerfabriken soll so auf 300 Verbrauchsorte verteilt werden, daß der Bedarf befriedigt wird und die Transportkosten minimiert werden.

Fabrik	$F_j \ (j = 1, \dots, 7)$,	Verbrauchsort	$G_k \ (k = 1, \dots, 300)$
Produktion	a_j (pro Monat) ,	Verbrauch	r_k (pro Monat)

transportierte Menge $F_j \to G_k \ : \ x_{j,k}$, Kosten $c_{j,k}$ (pro Einheit).

Es sei vorausgesetzt, dass Bedarf und Produktionsmenge gleich sind, d. h.:

$$\sum_{k=1}^{300} r_k = \sum_{j=1}^{7} a_j .$$

Zu minimieren sind die Gesamtkosten

$$Q(x_{1,1}, \ldots, x_{7,300}) := \sum_{j=1}^{7} \sum_{k=1}^{300} c_{j,k} x_{j,k}$$

unter den Nebenbedingungen $x_{j,k} \geq 0$ und

$$\sum_{j=1}^{7} x_{j,k} = r_k \quad (k = 1, \ldots, 300), \qquad \sum_{k=1}^{300} x_{j,k} = a_j \quad (j = 1, \ldots, 7).$$

Diese beiden Aufgabenstellungen lassen sich in folgenden allgemeinen Rahmen einordnen: (Für einen Vektor $x = (x_1, \ldots, x_n)^T$ bedeutet die Schreibweise $x \geq 0$, dass alle Komponenten $x_i \geq 0$ sind.) Für $1 \leq m \leq n$ seien eine Matrix $A \in \mathbb{R}^{m \times n}$ vom Rang m sowie Vektoren $b \in \mathbb{R}^m$, $b \geq 0$, und $c \in \mathbb{R}^n$ gegeben.

Definition 8.1: *Als „lineares Programm in Normalform" (oder auch „kanonischer" Form), abgekürzt „LP", bezeichnet man die Aufgabe, unter den Gleichungsnebenbedingungen und Vorzeichenbedingungen*

$$Ax = b, \quad x \geq 0$$

ein Minimum der Zielfunktion $Q(x) := c^T x$ zu bestimmen.

Anders ausgedrückt sucht ein lineares Programm im „zulässigen Bereich"

$$M := \{ x \in \mathbb{R}^n \mid Ax = b, \quad x \geq 0 \} \tag{8.1.2}$$

ein $x^* \in M$ zu bestimmen, so dass

$$c^T x^* = \min_{x \in M} c^T x. \tag{8.1.3}$$

Zur Einordnung der obigen Beispiele in diesen Rahmen können folgende Umformungen herangezogen werden:

- Ein Ungleichung mit $\geq$ wird durch Multiplikation mit -1 in eine mit $\leq$ überführt.
- Eine Ungleichung $a_1 x_1 + \ldots + a_n x_n \leq \beta$ wird durch Einführung einer sog. „Schlupfvariablen" y in eine Gleichung und eine Vorzeichenbedingung überführt:

$$a_1 x_1 + \ldots + a_n x_n + y = \beta, \quad y \geq 0.$$

- Für jede Gleichung $a_1 x_1 + \ldots + a_n x_n = \beta$ kann (eventuell nach Multiplikation mit -1) stets $\beta \geq 0$ vorausgesetzt werden.
- Fehlt für eine Variable, etwa für x_1, die Vorzeichenbedingung, so wird x_1 durch die Differenz $y_1 - y_2$ zweier neuer Variablen ersetzt, und man fordert $y_1 \geq 0$, $y_2 \geq 0$.

– Gleichungen, die Linearkombinationen anderer Gleichungen sind, werden weggelassen, so dass für die Matrix $A \in \mathbb{R}^{m\times n}$ stets Rang $A = m$ angenommen werden kann.

– Wegen $\max c^T x = -\min(-c^T x)$ kann man alle linearen Programme auf die Bestimmung eines Minimums zurückführen.

Der zulässige Bereich M eines LP ist der Durchschnitt einer linearen Mannigfaltigkeit mit Halbräumen und folglich abgeschlossen. Weiter ist M *„konvex“*:

$$x, y \in M \Longrightarrow \lambda x + (1-\lambda)\, y \in M \quad \forall\, \lambda \in [0,1].$$

Ist $M = \emptyset$, so besitzt das LP keine Lösung. Im Fall $M \neq \emptyset$ existiert immer eine Lösung, wenn M beschränkt (und damit kompakt) ist; für unbeschränktes M braucht keine Lösung zu existieren.

Beispiel 8.3: In einfachen Fällen lassen sich lineare Programme graphisch lösen: Der zulässige Bereich M in Beispiel 1 (Standardformulierung) ist der Durchschnitt von 5 Halbebenen des $\mathbb{R}^2$, deren Begrenzungsgeraden durch die Gleichungen

$$\begin{aligned} x_1 \geq 0, \quad x_2 &\geq 0, \\ x_1 + x_2 &= 100 \\ 4x_1 + x_2 &= 160 \\ 20x_1 + 10x_2 &= 1100 \end{aligned}$$

gegeben sind:

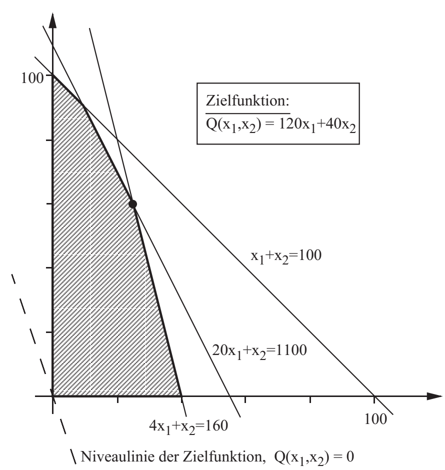

Abbildung 8.1: *Grafische Lösung eines Linearen Programms*

Parallelverschiebung der Niveaulinie bis an den Rand von M ergibt als maximalen Wert den der Niveaulinie durch den Punkt (x_1^*, x_2^*):

$$\left.\begin{array}{rcr} 4x_1^* + \quad x_2^* & = & 160 \\ 20x_1^* + 10x_2^* & = & 1100 \end{array}\right\} \Longrightarrow \begin{array}{rcl} x_2^* & = & 60 \\ x_1^* & = & 25 \end{array},$$

$$Q_{\max} = 120x_1^* + 40x_2^* \;=\; 5400.$$

Der optimale Punkt ist in diesem Fall eine Ecke des Polygongebiets M. Dies ist kein Zufall und wird sich als wesentlicher Punkt bei der Behandlung allgemeinerer Probleme dieses Typs erweisen. Die maximal mögliche Stückzahl von $x_1 + x_2 = 100$ wird unter dem Kriterium der Gewinnmaximierung also nicht erreicht; dafür wird die zur Verfügung stehende Arbeitszeit voll genutzt.

Wir betrachten im Folgenden die linearen Programmierungsaufgaben stets in Normalform. Zunächst studieren wir die Struktur der Lösungsmenge eines LP.

Definition 8.2: *Ein Vektor $x \in M$ heißt „Ecke“ (oder„Extremalpunkt“) der zulässigen Menge M, wenn er keine Darstellung*

$$x = \lambda x^{(1)} + (1-\lambda)x^{(2)}$$

mit $x^{(1)}$, $x^{(2)} \in M$, $x^{(1)} \neq x^{(2)}$, mit einem $\lambda \in (0,1)$ zulasst. Für $x \in M$ setzen wir $I(x) := \{i \in \{1,\dots,n\} \,|\, x_i > 0\}$. Weiter bezeichnen wir im Folgenden mit a_k die Spaltenvektoren der Matrix A.

Hilfssatz 8.1 (Sekantensatz): *Sind für ein $x \in M$ die Spaltenvektoren in*

$$B(x) := \{\, a_k \,|\, k \in I(x)\,\} \tag{8.1.4}$$

linear abhängig, so besitzt x eine Darstellung

$$x = \tfrac{1}{2}\,(x^{(1)} + y) \tag{8.1.5}$$

mit $x^{(1)}$, $y \in M$ und $I(x^{(1)}) \subsetneq I(x)$.

Beweis: O.B.d.A. sei $I(x) = \{1,\dots,k\}$, so dass

$$\sum_{i=1}^{k} x_i a_i = b.$$

Ist $B(x)$ linear abhängig, so gibt es Zahlen d_i, $i = 1,\dots,k$, nicht alle Null, so dass

$$\sum_{i=1}^{k} d_i a_i = 0.$$

Der Vektor

$$x(\lambda) = (x_1 + \lambda d_1, \dots, x_k + \lambda d_k, \underbrace{0,\dots,0}_{n-k})^T$$

erfüllt dann für jedes $\lambda \in \mathbb{R}$ die Gleichung $Ax(\lambda) = b$. Wegen $x_i > 0,\ i = 1, \ldots, k$, ist für hinreichend kleines $|\lambda|$ auch $x(\lambda) \geq 0$ und somit $x(\lambda) \in M$. Lässt man nun λ ausgehend von Null wachsen oder fallen, so gelangt man in einem der beiden Fälle zu einem λ^*, für das mindestens eine der Komponenten $x_i(\lambda^*)$, $i = 1, \ldots, k$, verschwindet. Ferner ist $x(\lambda) \in M$ für $|\lambda| \leq |\lambda^*|$. Mit $x^{(1)} = x(\lambda^*)$ und $y = x(-\lambda^*)$ gilt dann $x = \frac{1}{2}(x^{(1)} + y)$ und $I(x^{(1)}) \subsetneqq I(x)$. Q.E.D.

Hilfssatz 8.2 (Eckensatz): *Ein Vektor $x \in M$ ist genau dann Ecke, wenn die Spaltenvektoren in $B(x)$ linear unabhängig sind.*

Beweis: Aus Hilfssatz 8.1 folgt, dass für eine Ecke $x \in M$ notwendig $B(x)$ linear unabhängig sein muß. Sei nun $B(x)$ linear unabhängig, aber $x \in M$ keine Ecke, d. h. $x = \lambda x^{(1)} + (1-\lambda)x^{(2)}$ mit $x^{(1)}, x^{(2)} \in M, x^{(1)} \neq x^{(2)}, \lambda \in (0,1)$. Dann impliziert $x_i = 0$ auch $x_i^{(1)} = x_i^{(2)} = 0$, so daß gilt:

$$\sum_{i \in I(x)} x_i^{(1)} a_i = \sum_{i \in I(x)} x_i^{(2)} a_i = b.$$

Wegen $x^{(1)} \neq x^{(2)}$ ist also $B(x)$ linear abhängig im Widerspruch zur Annahme. Q.E.D.

Definition 8.3: *Wegen Rang $A = m$ besteht $B(x)$ für eine Ecke $x \in M$ aus höchstens m Vektoren. Ist für eine Ecke $x \in M$ aber* dim $B(x) < m$, *so heißt x „entartete Ecke"; in diesem Fall kann $B(x)$ zu einer Basis $\hat{B}(x)$ aus Spaltenvektoren von A ergänzt werden. In jedem Fall heißt eine solche Basis $\hat{B}(x)$ „Basis zur Ecke x ".*

Durch eine Basis ist eine Ecke $x \in M$ über das Gleichungssystem $Ax = b$ eindeutig bestimmt. Andererseits gibt es höchstens

$$\binom{n}{m} = \frac{n!}{m!(n-m)!}$$

Systeme von m linear unabhängigen Spaltenvektoren der Matrix A, d. h. Ecken von M.

Hilfssatz 8.3 (Eckenlösung): *Besitzt das LP eine Lösung $x \in M$, so gibt es eine Ecke $x^* \in M$, die ebenfalls Lösung ist.*

Beweis: Ist x selbst keine Ecke, so wenden wir Hilfssatz 8.1 an, d. h.: Das Minimum von $c^T x$ wird im Mittelpunkt $x = \frac{1}{2}(x^{(1)} + y)$ der Verbindungsgeraden zwischen $x^{(1)}$ und y angenommen. Folglich ist $c^T x$ dort konstant, d. h.: $x^{(1)}$ ist auch Lösung, aber mit $I(x^{(1)}) \subsetneqq I(x)$. Auf diese Weise erreicht man in endlich vielen Schritten eine Ecke von M (im Extremfall $\tilde{x} = 0$. Q.E.D.

8.2 Der Simplex-Algorithmus

Im Folgenden entwickeln wir den sog. „Simplex-Algorithmus“ nach G. B. Dantzig[1] (1947) zur Lösung von Linearen Programmen. Wir verwenden weiter die Bezeichnungen des vorherigen Abschnitts. Sei x^0 eine Ecke des zulässigen Bereichs M der kanonischen Programmierungsaufgabe

$$Ax = b\,, \qquad x \geq 0\,, \qquad c^T x = \min!$$

mit zugehöriger Basis $\hat{B}(x^0) = \{\, a_i\,,\ i \in I^0\,\}\,, I^0 \supseteq I(x^0)$. Dann gilt

$$\sum_{i\in I^0} x_i^0\, a_i = b. \tag{8.2.6}$$

Für ein beliebiges $x \in M$ ist $Ax = b$ und folglich

$$\sum_{i\in I^0}\{x_i - x_i^0\}a_i = -\sum_{i\notin I^0} x_i a_i.$$

(Die Schreibweise $i \notin I^0$ bedeutet $i \in \{1,\dots,n\} \setminus I^0$.) Wegen der linearen Unabhängigkeit von $\hat{B}(x^0)$ kann nach den Differenzen $x_i - x_i^0$ aufgelöst werden, und man erhält Gleichungen der Form

$$x_i = \sum_{k\notin I^0} \alpha_{ik} x_k + x_i^0\,, \quad i \in I^0. \tag{8.2.7}$$

Der zugehörige Zielfunktionswert

$$c^T x = c^T x^0 + c^T(x - x^0) = c^T x^0 + \sum_{i\in I^0} c_i(x_i - x_i^0) + \sum_{i\notin I^0} c_i x_i.$$

ergibt sich nach Substitution von $x_i - x_i^0\ (i \in I^0)$ in der Form

$$c^T x = \sum_{k\notin I^0} \gamma_k x_k + c^T x^0 \tag{8.2.8}$$

$$\gamma_k = \sum_{i\in I^0} \alpha_{ik} c_i + c_k\,, \quad k \notin I^0. \tag{8.2.9}$$

Die Gleichungsnebenbedingung und die Zielfunktion $c^T x$ sind offenbar (bzgl. der Basis $\hat{B}(x^0)$) durch das folgende $(m+1) \times (n-m+1)$-Gleichungssystem wiedergegeben:

[1] George B. Dantzig (1914–2005): US-Amerikanischer Mathematiker; entwickelte 1947 den Simplex-Algorithmus wärend seiner Tätigkeit in einem Forschungslabor der U.S. Air Force; s. sein Buch „Lineare Programmierung und Erweiterungen“, Springer 1966 (Übersetzung aus dem Englischen); seit 1966 Professor an der Stanford University.

$$\begin{aligned} x_i &= \sum_{k \notin I^0} \alpha_{ik} x_k + x_i^0, \quad i \in I^0 \\ c^T x &= \sum_{k \notin I^0} \gamma_k x_k + c^T x^0. \end{aligned} \tag{8.2.10}$$

Die Komponenten x_i, $i \in I^0$, und der zugehörige Zielfunktionswert z eines Vektors $x \in \mathbb{R}^n$ sind durch Vorgabe von $x_k \geq 0$ $(k \notin I^0)$ in (8.2.10) eindeutig bestimmt. Gilt dabei $x_i \geq 0\,(i \in I^0)$, so ist $x \in M$. Für die speziellen Werte $x_k = 0\,(k \notin I^0)$ ergibt sich gerade die Ausgangsecke x^0.

Wir betrachten nun die umgekehrte Situation, dass der zulässige Bereich M gerade aus denjenigen Vektoren $x \geq 0$ besteht, deren Komponenten einem System der Gestalt (8.2.10) genügen, mit gewissen Zahlen $x_i^0 \geq 0\,(i \in I^0)$. Dann ist der Vektor $x^0 \in \mathbb{R}^n$ mit den Komponenten $x_i^0\,(i \in I^0)$, $x_i^0 = 0\,(i \notin I^0)$ automatisch Ecke von M, denn er erfüllt offensichtlich (8.2.10), d. h.: $Ax^0 = b$, und jede Darstellung $x^0 = \lambda x + (1-\lambda)\tilde{x}$ mit $x, \tilde{x} \in M$, $0 < \lambda < 1$ impliziert zunächst notwendig $x_k = \tilde{x}_k = x_k^0\,(k \notin I^0)$ und damit auch $x_i = \tilde{x}_i = x_i^0\,(i \in I^0)$.

Das System (8.2.10) charakterisiert also zu einer festen Ecke x^0 bzw. der zugehörigen Basis $\hat{B}(x^0)$ den zulässigen Bereich M. Der Simplex-Algorithmus sucht nun eine (zu x^0 benachbarte) Ecke x^1 von M, wobei möglichst $c^T x^1 < c^T x^0$ gelten soll. Der zugehörige Basiswechsel in der Darstellung (8.2.10) wird mit Hilfe des schon betrachteten „Gauß-Jordan-Algorithmus" bewerkstelligt (s. Kapitel 4.2). Der Vollständigkeit halber rekapitulieren wir im Folgenden den Gauß-Jordan-Algorithmus. Dieser dient zur Lösung linearer Gleichungssysteme $Ax = y$ mit $A \in \mathbb{R}^{m\times n}$, $x \in \mathbb{R}^n$, $y \in \mathbb{R}^m$, durch sukzessiven Austausch der Komponenten von x gegen solche von y. Ist ein Matrixelement $a_{pq} \neq 0$, so kann die p-te Gleichung nach x_q aufgelöst werden:

$$x_q = -\frac{a_{p1}}{a_{pq}} x_1 - \ldots - \frac{a_{p,q-1}}{a_{pq}} x_{q-1} + \frac{1}{a_{pq}} y_p - \frac{a_{p,q+1}}{a_{pq}} x_{q+1} - \ldots - \frac{a_{pn}}{a_{pq}} x_n\,.$$

Durch Substitution von x_q in den anderen Gleichungen

$$a_{j1}x_1 + \ldots + a_{j,q-1}x_{q-1} + a_{jq}\boxed{x_q} + a_{j,q+1}x_{q+1} + \ldots + a_{jn}x_n = y_j$$

erhält man für $j = 1, \ldots, m,\ j \neq p$:

$$\begin{aligned} &\left[a_{j1} - \frac{a_{jq}a_{p1}}{a_{pq}} \right] x_1 + \ldots + \left[a_{j,q-1} - \frac{a_{jq}a_{p,q-1}}{a_{pq}} \right] x_{q-1} + \frac{a_{jq}}{a_{pq}} y_p + \\ &+ \left[a_{j,q+1} - \frac{a_{jq}a_{p,q+1}}{a_{pq}} \right] x_{q+1} + \ldots + \left[a_{jn} - \frac{a_{jq}a_{pn}}{a_{pq}} \right] x_n = y_j\,. \end{aligned}$$

Das Resultat ist ein zum Ausgangssystem äquivalentes System

$$\tilde{A}\begin{bmatrix} x_1 \\ \vdots \\ y_p \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} y_1 \\ \vdots \\ x_q \\ \vdots \\ y_m \end{bmatrix}, \tag{8.2.11}$$

wobei die Elemente der Matrix $\tilde{A}$ wie folgt bestimmt sind:

- Pivotelement: $\tilde{a}_{pq} = \frac{1}{a_{pq}}$,
- Pivotzeile: $\tilde{a}_{pk} = -\frac{a_{pk}}{a_{pq}}, \quad k = 1, \dots, n, \quad k \neq q\,,$
- Pivotspalte: $\tilde{a}_{jq} = \frac{a_{jq}}{a_{pq}}\,, \quad j = 1, \dots, m, \quad j \neq p\,,$
- sonstige: $\tilde{a}_{jk} = a_{jk} - a_{jq}\frac{a_{pk}}{a_{pq}}, \quad \begin{matrix} j = 1, \dots, m, & j \neq p \\ k = 1, \dots, n, & k \neq q. \end{matrix}$

Gelingt es, durch Fortsetzung des Verfahrens alle Komponenten von x durch solche von y zu ersetzen, so hat man eine explizite Darstellung der Lösung von $Ax = y$. Im Fall $m = n$ ergibt sich so auch die Inverse A^{-1}, allerdings i. Allg. mit vertauschten Zeilen und Spalten. Bei der Festlegung des Pivotelementes empfiehlt es sich aus Stabilitätsgründen, unter allen in Frage kommenden a_{pq} jeweils eines von möglichst großem Betrag zu wählen. Für ein quadratisches Gleichungssystem mit regulärer Koeffizientenmatrix A ist das Gauß-Jordan-Verfahren zur Berechnung von A^{-1} stets durchführbar.

Beispiel 8.4:

$$\begin{bmatrix} 1 & 2 & 1 \\ -3 & -5 & -1 \\ -7 & -12 & -2 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}$$

Austauschschritte: $\boxed{\cdot}$ Pivotelement

$$\begin{array}{rrr|l} x_1 & x_2 & x_3 & \\ \hline 1 & 2 & 1 & y_1 \\ -3 & -5 & -1 & y_2 \\ -7 & \boxed{-12} & -2 & y_3 \end{array} \qquad \begin{array}{rrr|l} x_1 & y_3 & x_3 & \\ \hline -1/6 & -1/6 & \boxed{2/3} & y_1 \\ -1/12 & 5/12 & -5/6 & y_2 \\ -7/12 & -1/12 & -1/6 & x_2 \end{array}$$

$$\begin{array}{rrr|l} x_1 & y_3 & y_1 & \\ \hline 1/4 & 1/4 & 3/2 & x_3 \\ \boxed{-1/8} & 3/8 & -1/4 & y_2 \\ -5/8 & -1/8 & -1/4 & x_2 \end{array} \qquad \begin{array}{rrr|l} y_2 & y_3 & y_1 & \\ \hline -2 & 1 & 1 & x_3 \\ -8 & 3 & -2 & x_1 \\ 5 & -2 & 1 & x_2 \end{array}$$

$$\text{Inverse:} \begin{bmatrix} -2 & -8 & 3 \\ 1 & 5 & -2 \\ 1 & -2 & 1 \end{bmatrix}.$$

Der Simplex-Algorithmus besteht aus zwei „Phasen“. In Phase I wird eine Ausgangsecke x^0 konstruiert und das zugehörige *Tableau* erstellt:

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

In Phase II werden dann unter Verwendung des Gauß-Jordan-Algorithmus Basiswechsel vollzogen, wobei der Zielfunktionswert jeweils möglichst stark verkleinert wird. Wir beginnen mit der Beschreibung dieses Basisaustausches.

Phase II (Basisaustausch)

1. Gilt $\gamma_i \geq 0\ (i \notin I^0)$, so folgt für beliebige Vorgabe von $x_i \geq 0\,(i \notin I^0)$, d. h.: für beliebige Punkte $x \in M$, stets

$$z = \sum_{k \notin I^0} \gamma_k x_k + c^T x^0 \geq c^T x^0,$$

d. h. Die Startecke x^0 ist bereits optimal.

2. Gibt es ein $q \notin I^0$ mit $\gamma_q < 0$, so sind zwei Fälle zu unterscheiden:

   (a) Im Falle $\alpha_{iq} \geq 0\,(i \in I^0)$ erhält man durch Vorgabe von $x_q := \lambda$, $\lambda \in \mathbb{R}_+$ und $x_k = 0\ (k \notin I^0,\ k \neq q)$ Vektoren $x \in M$,

$$x_i = \alpha_{iq}\lambda + x_i^0 \geq 0,$$

   mit Zielfunktionswert

$$z = \gamma_q \lambda + c^T x^0 \to -\infty \quad (\lambda \to \infty).$$

   Die Aufgabe ist also unlösbar.

   (b) Gibt es Indizes $p \in I^0$ mit $\alpha_{pq} < 0$, so wird die Variable x_q gegen eine noch auszuwählende x_p ausgetauscht. Die Elemente des Tableaus sind dabei gemäß dem Gauß-Jordan-Algorithmus wie folgt zu transformieren:

   Pivotelement: $\alpha_{pq} \to \alpha'_{qp} = \dfrac{1}{\alpha_{pq}}$;

$$
\begin{array}{ll}
\text{Pivotspalte:} & \alpha_{iq} \to \alpha'_{ip} = \dfrac{\alpha_{iq}}{\alpha_{pq}} \quad (i \in I^0 \setminus \{p\}); \\[2ex]
(\gamma_q \to \gamma_p) & \gamma_q \to \gamma'_p = \dfrac{\gamma_q}{\alpha_{pq}}; \\[2ex]
\text{Pivotzeile:} & \alpha_{pk} \to \alpha'_{qk} = -\dfrac{\alpha_{pk}}{\alpha_{pq}} \quad (k \notin I^0,\ k \neq q); \\[2ex]
(x_p \to x_q) & x_p^0 \to x_q^1 = -\dfrac{x_p^0}{\alpha_{pq}}\,; \\[2ex]
\text{sonstige:} & \alpha_{ik} \to \alpha'_{ik} = \alpha_{ik} - \dfrac{\alpha_{iq}\alpha_{pk}}{\alpha_{pq}}; \\[2ex]
(x_k \to x_k,\ x_i \to x_i) & x_i^0 \to x_i^1 = x_i^0 - \dfrac{\alpha_{iq}x_p^0}{\alpha_{pq}}, \qquad \gamma_k \to \gamma'_k = \gamma_k - \dfrac{\gamma_q\alpha_{pk}}{\alpha_{pq}}; \\[2ex]
 & c^Tx^0 \to c^Tx^1 = c^Tx^0 - \dfrac{\gamma_q x_p^0}{\alpha_{pq}}\,.
\end{array}
$$

Auswahlregel (R): *Der Index $p \in I^0$ wird dabei gemäß der folgenden Regel ausgewählt:*

$$
\alpha_{pq} < 0\,, \quad \frac{x_p^0}{\alpha_{pq}} = \max_{\alpha_{iq}<0, i\in I^0} \frac{x_i^0}{\alpha_{iq}}. \tag{8.2.12}
$$

Satz 8.1 (Simplex-Algorithmus): *Wird der Basisaustausch gemäß der Regel* (R) *vorgenommen, so ist der Vektor $x^1 \in \mathbb{R}^n$ mit den Komponenten $x_i^1 > 0$ $(i \in I^1 := [I^0 \setminus \{p\}] \cup \{q\})$, $x_i^1 = 0$ $(i \notin I^1)$ wieder eine Ecke von M mit der zugehörigen Basis*

$$
\hat{B}(x^1) = \Big[\hat{B}(x^0) \setminus \{a_p\}\Big] \cup \{a_q\}, \tag{8.2.13}
$$

und es gilt $c^Tx^1 \le c^Tx^0$. Im Falle $x_p^0 > 0$ ist sogar $c^Tx^1 < c^Tx^0$.

Beweis: Wegen $\alpha_{pq} < 0$ folgt $x_q^1 = -x_p^0/\alpha_{pq} \ge 0$. Ist weiter $\alpha_{iq} \ge 0$, so folgt $x_i^1 = x_i^0 - \alpha_{iq}x_p^0/\alpha_{pq} \ge 0$. Im Falle $\alpha_{iq} < 0$ gilt ebenfalls wegen der Auswahlregel (R):

$$
\frac{x_i^1}{\alpha_{iq}} = \frac{x_i^0}{\alpha_{iq}} - \frac{x_p^0}{\alpha_{pq}} \le 0 \qquad \Longrightarrow \qquad x_i^1 \ge 0\,..
$$

Nach dem oben gesagten ist x^1 also Ecke von M. Q.E.D.

Der Eckenaustausch nach (2b) kann solange fortgesetzt werden, bis Fall (1) oder Fall (2a) eintritt. Eine *nicht* entartete Ecke kann dabei nie ein zweites Mal erreicht werden, da ihr Austausch je zu einer Verkleinerung des Zielfunktionswertes führt. Das Auftreten entarteter Ecken wird weiter unten diskutiert werden. Hier könnten sich (theoretisch) im Laufe des Verfahrens verschiedene Basen zu einer entarteten Ecke zyklisch wiederholen, so dass der Algorithmus nicht abbricht.

Beispiel 8.5: Beispiel 8.1.1 aus Abschnitt 8.1 erhält nach Einführung von Schlupfvariablen x_3, x_4, x_5 die Form

$$-120x_1 - 40x_2 = \min!\,, \quad x_i \geq 0\,, \quad i = 1, \ldots, 5,$$

$$\begin{array}{rcrcccccccr} x_1 & + & x_2 & + & x_3 & & & & & = & 100 \\ 4x_1 & + & x_2 & + & & + & x_4 & & & = & 160 \\ 20x_1 & + & 10x_2 & & & & & + & x_5 & = & 1100 \end{array}$$

Offensichtlich ist $x^0 = (0, 0, 100, 160, 1100)^T$ eine *nicht* entartete Ecke mit der Basis

$$B(x^0) = \{a_3, a_4, a_5\} = \left\{ \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \right\}.$$

Das Ausgangstableau ist also:

I	x_1	x_2	$x_6 = 1$	x_i^0/α_{iq}
x_3	-1	-1	100	-100
x_4	$\boxed{-4}$	-1	160	-40
x_5	-20	-10	1100	-55
z	$\boxed{-120}$	-40	0	

$\boxed{\text{Fall (2b)}}$

Wahl $q = 1$ (oder $q = 2$)

Regel (R) $\Longrightarrow p = 4$.

Eckentausch (Pivotelement α_{41})

II	x_4	x_2	$x_6 = 1$	x_i^0/α_{iq}
x_3	$1/4$	$-3/4$	60	-80
x_1	$-1/4$	$-1/4$	40	-160
x_5	5	$\boxed{-5}$	300	-60
z	30	$\boxed{-10}$	-4800	

$\boxed{\text{Fall (2b)}}$

Wahl $q = 2$

Regel (R) $\Longrightarrow p = 5$.

Eckentausch (Pivotelement α_{52})

III	x_4	x_5	$x_6 = 1$
x_3	$-1/2$	$-3/20$	15
x_1	$-1/2$	$1/20$	25
x_2	1	$-1/5$	60
z	20	2	-5400

$\boxed{\text{Fall (1)}}$

Eckenlösung $x^3 = (25, 60, 15, 0, 0)^T$

Extremwert $z = -5400$.

Wie wir schon gesehen haben, wird der maximale Gewinn von 5400 DM erreicht, wenn 25 Produkte des Typs A und 60 des Typs B hergestellt werden.

Bemerkung 8.1: Zur Kontrolle der Rechnung sollten die Größen $\gamma_k\,(k \notin I^0)$ zusätzlich auch aus der folgenden Formel berechnet werden:

$$\gamma_k = \sum_{i \in I^0} \alpha_{ik} c_i + c_k. \tag{8.2.14}$$

Phase I (Konstruktion einer Startecke)

Wir diskutieren nun die Phase I des Simplexalgorithmus, d. h. die Konstruktion einer Ausgangsecke x^0. Im Falle eines in Standardform gegebenen Programms mit $b \geq 0$ ist dies, wie obiges Beispiel zeigt, sehr einfach:

$$\text{(I)} \qquad c^T x = \max!\,, \quad Ax \leq b\,, \quad x \geq 0. \tag{8.2.15}$$

Durch Einführung von Schlupfvariablen $v \in \mathbb{R}^m$ geht (I) in die kanonische Form über:

$$(\tilde{\text{I}}) \qquad \tilde{c}^T \tilde{x} = \min!\,, \quad \tilde{A}\tilde{x} = \tilde{b}\,, \quad \tilde{x} \geq 0, \tag{8.2.16}$$

mit

$$\tilde{A} = [A, I_m]\,, \quad \tilde{b} = b\,, \quad \tilde{c} = (-c, 0_m)^T, \quad \tilde{x} = (x, v)^T.$$

Wegen $b \geq 0$ ist der Vektor $\tilde{x}^0 = (0_n, \tilde{b}) \in \mathbb{R}^{n+m}$ automatisch eine Ecke von $\tilde{M}$, denn die zugehörigen Spaltenvektoren von $\tilde{A}$ bilden gerade die Einheitsmatrix I_m.

Ist die Programmierungsaufgabe in kanonischer Form gestellt,

$$\text{(II)} \qquad c^T x = \min!\,, \quad Ax = b\,, \quad x \geq 0, \tag{8.2.17}$$

oder ist in (I) $b \geq 0$ nicht erfüllt, so ist i. Allg. keine Ecke von M (oft nicht einmal ein zulässiger Vektor) ersichtlich. Zu ihrer Konstruktion betrachte man das Hilfsproblem (o.B.d.A.: $\tilde{b} \geq 0$)

$$(\tilde{\text{II}}) \qquad \tilde{c}^T \tilde{x} = \min!\,, \quad \tilde{A}\tilde{x} = \tilde{b}\,, \quad \tilde{x} \geq 0, \tag{8.2.18}$$

mit $v \in \mathbb{R}^m$,

$$\tilde{A} = [A, I_m]\,, \quad \tilde{b} = b\,, \quad \tilde{c} = (0_n, 1, \ldots, 1)^T\,, \quad \tilde{x} = (x, v)^T.$$

Hier ist mit $\tilde{x}^0 = (0_n, \tilde{b})$ eine Ausgangsecke von $\tilde{M}$ bekannt. Wegen $\tilde{x} \geq 0$ auf M besitzt das Hilfsproblem stets eine Lösung $\tilde{x}^* = (x^*, v^*)$, die man mit Hilfe des Simplex-Algorithmus bestimmen kann. Ist $v^* = 0$, so liefern die ersten n Komponenten der Lösung $\tilde{x}^*$ eine Ausgangsecke des ursprünglichen LP:

$$x_i^0 := \tilde{x}_i^*\,, \quad i = 1, \ldots, n. \tag{8.2.19}$$

Im Fall $v_i^* > 0$ für ein i muß $M = \emptyset$ sein, d. h.: Das LP besitzt keine Lösung. Mit dem Simplex-Algorithmus kann also auch die Frage nach der Lösbarkeit des LP entschieden werden.

Nach den bisherigen Überlegungen ist der Simplex-Algorithmus (mit der Auswahlregel (R)) grundsätzlich geeignet zur Lösung linearer Programmierungsaufgaben, bzw. zur Entscheidung ihrer Unlösbarkeit, vorausgesetzt, es treten keine entarteten Ecken auf. Das Erscheinen einer entarteten Ecke x^1 ist dadurch gekennzeichnet, dass im vorangehenden Austauschschritt das Kriterium (R) nicht zu einem eindeutigen Index $p \in I^0$ führt:

$$x^1 \quad \textit{nicht}\text{ entartet} \iff \begin{array}{c} \exists!\, p \in I^0 : \alpha_{pq} < 0 \\ \frac{x_p^0}{\alpha_{pq}} = \max\limits_{\alpha_{iq}<0} \frac{x_i^0}{\alpha_{iq}}; \end{array}$$

andernfalls folgte mit $x_p^0/\alpha_{pq} = x_{p'}^0/\alpha_{p'q}$

$$\left.\begin{array}{rcccc} x_p^1 & = & x_p^0 - \alpha_{pq}x_p^0/\alpha_{pq} & = & 0 \\ x_{p'}^1 & = & x_{p'}^0 - \alpha_{p'q}x_p^0/\alpha_{pq} & = & 0 \end{array}\right\} \iff \begin{array}{l} x^1 \text{ hat weniger als } m \\ \text{positive Komponenten.} \end{array}$$

Tritt im Verlaufe des Simplexverfahrens eine entartete Ecke x^0 auf, so kann es passieren, daß für den Index $p \in I^0$ gerade $x_p^0 = 0$ ist. Dann bewirkt der Austauschschritt offenbar keine Veränderung des Vektors x^0, insbesondere also keine Reduzierung des Zielfunktionswertes, sonderen nur den Übergang zu einer anderen Basis zur Ecke x^0. Wiederholt sich dann dieselbe Basis zyklisch, so führt das Verfahren nicht zum Ziel. Obwohl in der Praxis häufig entartete Ecken auftreten, sind derartige Zyklen noch nicht beobachtet worden (nur bei eigens zu diesem Zweck konstruierten pathologischen Beispielen). Für Belange der Praxis erscheint der Simplex-Algorithmus mit der Auswahlregel (R) (ergänzt um eine geeignete Regel für den Fall einer entarteten Ecke) als hinreichend robust. Vom theoretischen Standpunkt ist diese Situation aber unbefriedigend, und man sucht nach einer Auswahlregel, mit der der Algorithmus garantiert zum Ziel führt.

Definition 8.4 (Lexikographische Ordnung): *Ein Vektor $u \in \mathbb{R}^n$ heißt „lexikographisch positiv", $u \overset{\rightarrow}{>} 0$, wenn $u \neq 0$ ist und die erste nicht verschwindende Komponente positiv ist. Ein Vektor $u \in \mathbb{R}^n$ heißt „lexikographisch kleiner (größer)" als ein $v \in \mathbb{R}^n$, wenn $v-u \overset{\rightarrow}{>} 0$ $(u-v \overset{\rightarrow}{>} 0)$. Damit ist auf dem $\mathbb{R}^n$ eine "Ordnung" erklärt.*

Der Simplex-Algorithmus sei gestartet mit einer Ecke x^{start} mit der Basis $\hat{B}(x^{\text{start}}) = \{a_1, \dots, a_m\}$ (gegebenenfalls nach Umbenennung der Variablen). Damit ist $I^{\text{start}} = \{1, \dots, m\}$. Zur Einführung einer erweiterten Auswahlregel werden die Parameter α_{ik} und γ_k auch für $k \in I^{\text{start}}$ erklärt durch

$$\alpha_{ik} := -\delta_{ik}, \quad \gamma_k := 0, \quad i, k \in I^{\text{start}},$$

Hilfssatz 8.4: *Für die durch die Darstellungen*

$$a_i = \sum_{k \in I^0} c_{ik} a_k, \quad i \in \{1, \dots, n\} \tag{8.2.20}$$

eindeutig bestimmten Zahlen c_{ik} gilt:

$$c_{ki} = -\alpha_{ik}, \quad k \in \{1, \dots, n\}, \quad i \in I^{\text{start}}. \tag{8.2.21}$$

Beweis: Für $i,k \in I^{\text{start}}$ ist nach Definition

$$\alpha_{ik} = -\delta_{ki} = -c_{ki}.$$

Sei nun $x \in M$ beliebig. Dann gilt

$$\sum_{i\in I^{\text{start}}} \{x_i - x_i^0\} a_i = -\sum_{i\notin I^{\text{start}}} x_i a_i = -\sum_{i\notin I^{\text{start}}} x_i \sum_{k\in I^{\text{start}}} c_{ik} a_k = -\sum_{i\in I^{\text{start}}} (\sum_{k\notin I^{\text{start}}} c_{ki} x_k) a_i$$

und folglich wegen der linearen Unabhängigkeit von $\hat{B}(x^0)$

$$x_i - x_i^0 = -\sum_{k\notin I^{\text{start}}} c_{ki} x_k\,, \quad i \in I^{\text{start}}.$$

Da die α_{ik} in der Darstellung (8.2.10) eindeutig bestimmt sind, ergibt sich notwendig $c_{ki} = -\alpha_{ik}\ (k \in I^{\text{start}})$. Q.E.D.

Sei nun x^0 eine im Verlaufe des Verfahrens erreichte Ecke und $q \in I^0$ der Austauschindex. Zur Bestimmung des Index $p \in I^0$ bilde man für alle $r \in I^0$ mit

$$\frac{x_r^0}{\alpha_{rq}} = \max_{\alpha_{jq}<0} \frac{x_j^0}{\alpha_{jq}}, \quad \alpha_{rq} < 0, \tag{8.2.22}$$

die Vektoren

$$u^r = \left(\frac{x_r^0}{\alpha_{rq}}, -\frac{\alpha_{r1}}{\alpha_{rq}}, \dots, -\frac{\alpha_{rm}}{\alpha_{rq}}\right)^T \in \mathbb{R}^{m+1}.$$

Auswahlregel ($\tilde{\text{R}}$): *Der Index $p \in I^0$ wird dann als derjenige mit der Eigenschaft (8.2.22) gewählt, so dass u^p der lexikographisch größte unter den u^r ist.*

Im Falle, dass der „maximale“ Index in (8.2.22) eindeutig bestimmt ist, stimmt die Auswahlregel ($\tilde{\text{R}}$) offenbar mit (R) überein. Ansonsten ist durch ($\tilde{\text{R}}$) eindeutig ein $p \in I^0$ festgelegt; denn gäbe es keines, so wären für zwei $p, p' \in I^0$ die Vektoren u^p, $u^{p'}$ identisch. Dies bedeutete aber, dass die quadratische Matrix $(\alpha_{ik})_{i\in I^0, k=1,\dots,m}$ zwei zueinander proportionale Zeilen hätte und somit singulär wäre. Nach Hilfssatz 8.4 wäre dann auch $(c_{ik})_{i=1,\dots,m,k\in I^0}$ singulär im Widerspruch zur linearen Unabhängigkeit der Vektoren in $\hat{B}(x^0)$ und $\hat{B}(x^{\text{start}})$.

Der Ecke x^0 ordnen wir nun den folgenden Vektor zu:

$$v^0 = (c^T x^0, c_1 - \gamma_1, \dots, c_m - \gamma_m)^T \in \mathbb{R}^{m+1}$$

Hilfssatz 8.5 (Reduktionssatz): *Beim Eckenaustausch $x^0 \to x^1$ unter der Bedingung der Auswahlvorschrift ($\tilde{\text{R}}$) wird der Vektor v^0 durch einen lexikographisch kleineren Vektoren v^1 ersetzt.*

Beweis: Die Indexmenge I^0 wird ersetzt durch $I^1 = (I^0 \setminus \{p\}) \cup \{q\}$. Die γ_i werden nach den folgenden Regeln transformiert:

$$\begin{aligned}
&k \in I^0\,, \quad k \neq q\,: && \gamma_k \to \gamma_k - \gamma_q\,\tfrac{\alpha_{pk}}{\alpha_{pq}}\\
& \quad k = q\,: && \gamma_q \to 0 \;=\; \gamma_q - \gamma_q\,\tfrac{\alpha_{pq}}{\alpha_{pq}}\\
&k \notin I^0\,, \quad k \neq p\,: && \gamma_k \to 0\\
& \quad k = p\,: && \gamma_p \to \gamma_q\,\tfrac{1}{\alpha_{pq}} = \gamma_p - \gamma_q\,\tfrac{\alpha_{pp}}{\alpha_{pq}} \quad (\gamma_p = 0\,,\ \alpha_{pp} = -1).
\end{aligned}$$

Ferner gilt: $\qquad c^T x^0 \to c^T x^0 - \gamma_k\,\frac{x_p^0}{\alpha_{pq}}\,.$

Für den zur neuen Ecke x^1 gehörenden Vektor $v^1 \in \mathbb{R}^{m+1}$ gilt also:

$$v^1 = v^0 - \gamma_q \begin{bmatrix} x_p^0/\alpha_{pq} \\ -\alpha_{p1}/\alpha_{pq} \\ \vdots \\ -\alpha_{pm}/\alpha_{pq} \end{bmatrix} = v^0 - \gamma_q u^p.$$

Da $\gamma_q < 0\,,\ \alpha_{pq} < 0$ bleibt zu zeigen, dass der Vektor $w^p = (x_p^0, -\alpha_{p1}, \dots, -\alpha_{pm})^T \in \mathbb{R}^{m+1}$ lexikographisch positiv ist. Dies geschieht durch Induktion bzgl. der Zahl der durchgeführten Verfahrensschritte.

(i) Die zur Ausgangsecke x^{start} gehörenden Vektoren $w^k\,(k = 1, \dots, m)$ sind trivialerweise lexikographisch positiv, denn es ist $x_k^0 \geq 0$ und $-\alpha_{ki} = \delta_{ki}\ (i = 1, \dots, m)\,.$

(ii) Sei x^0 eine im Verlaufe des Verfahrens auftretende Ecke, und alle zu x^0 gebildeten Vektoren $w^k\ (k \in I^0)$ seien lexikographisch positiv. Beim Übergang von x^0 zur Ecke x^1 ergeben sich die zugehörigen Vektoren $\tilde{w}^k\ (k \in I^1)$ wie folgt:

$$\begin{aligned}
k \in I^1\,,\quad k \neq q\,: \qquad w^k \;&=\; \Big(x_k^0 - \tfrac{\alpha_{kq} x_p^0}{\alpha_{pq}}\,, -\alpha_{k1} + \tfrac{\alpha_{kq}\alpha_{p1}}{\alpha_{pq}}\,, \dots, -\alpha_{km} + \tfrac{\alpha_{kq}\alpha_{pm}}{\alpha_{pq}} \Big)^T\\
&=\; w^k - \tfrac{\alpha_{kq}}{\alpha_{pq}}\, w^p\\
k = q\,: \qquad \tilde{w}^q \;&=\; \Big(\tfrac{x_p^0}{\alpha_{pq}}\,, +\tfrac{\alpha_{p1}}{\alpha_{pq}}\,, \dots, +\tfrac{\alpha_{pm}}{\alpha_{pq}} \Big)^T \;=\; -\tfrac{1}{\alpha_{pq}}\, w^p.
\end{aligned}$$

Hieraus entnehmen wir mit der Induktionsannahme:

$$\begin{aligned}
k \in I^1\,,\quad k \neq q\,:\quad &\text{a)} \quad \alpha_{kq} \geq 0 \Longrightarrow \tilde{w}^k = w^k + \left|\tfrac{\alpha_{kq}}{\alpha_{pq}}\right|\, w^p \overset{\rightarrow}{>} 0\,,\\
&\text{b)} \quad \alpha_{kq} < 0 \Longrightarrow \text{Auswahlregel } (\tilde{R})\ :\ u^p \overset{\rightarrow}{>} u^k\\
&\qquad\quad \Longrightarrow \tilde{w}^k = \alpha_{kq} u^k - \tfrac{\alpha_{kq}}{\alpha_{pq}} \alpha_{pq} u^p \overset{\rightarrow}{>} 0\,,\\
k = q\,:\quad &\qquad \tilde{w}^q = \left|\tfrac{1}{\alpha_{pq}}\right|\, w^p \overset{\rightarrow}{>} 0\,.
\end{aligned}$$

Dies vervollständigt den Beweis. Q.E.D.

Wir fassen die Ergebnisse der bisherigen Überlegungen zusammen:

Satz 8.2 (Erweiterter Simplex-Algorithmus): *Unter der obigen Voraussetzung an die Ausgangsecke liefert der Simplex-Algorithmus mit der Auswahlregel* $(\tilde{\mathrm{R}})$ *in endlich vielen Schritten eine Lösung des kanonischen Problems* (II) *oder die Bestätigung seiner Unlösbarkeit.*

Beweis: Nach Hilfssatz 8.5 kann aufgrund der Auswahlregel $(\tilde{\mathrm{R}})$ keine Basis von Spaltenvektoren von A zweimal auftreten, denn durch die Ecke x^0 und eine zugehörige Basis $\hat{B}(x^0)$ ist der Vektor v^0 eindeutig bestimmt. Zyklen werden also vermieden. Q.E.D.

8.3 Übungsaufgaben

Übung 8.1: Man bringe die folgenden Optimierungsaufgaben in die kanonische Form eines „linearen Programms“:

a)
$$\begin{aligned} Q(x) := x_1 + x_2 + x_3 &\to \min! \\ x_1 \geq 0, \quad x_2 \geq 0, \quad x_1 + 2x_2 &\leq 5, \\ x_2 + x_3 &\leq 0, \\ 3x_2 - 4x_3 &\leq 1. \end{aligned}$$

b)
$$\begin{aligned} Q(x) := |x_1| + |x_2| + |x_3| &\to \min! \\ x_1 + x_2 &\leq 1, \\ 2x_1 + x_3 &\leq 3. \end{aligned}$$

Übung 8.2: Man löse die folgende Optimierungsaufgabe grafisch:

$$\begin{aligned} Q(x) := 2x_1 + x_2 &\to \min! \\ x_1 \geq 0, \quad x_2 \geq 0, \quad -2x_1 + x_2 &\leq -2, \\ x_1 - 2x_2 &\leq 2, \\ -x_1 - x_2 &\leq -5. \end{aligned}$$

Ist die Aufgabe lösbar, wenn $Q(x) \to \max!$ gefordert wird?

Übung 8.3: a) Man wende den Gauß-Jordan-Algorithmus auf die folgende 3×3-Matrix an:

$$A = \begin{pmatrix} 1 & 0 & -1 \\ 2 & 2 & 0 \\ 1 & 2 & 1 \end{pmatrix}.$$

Mit Hilfe des Ergebnisses bestimme man alle $y \in \mathbb{R}^3$, für die das Gleichungssystem $Ax = y$ eine Lösung besitzt, sowie die jeweilige Lösungsmenge.

b) Man implementiere den Gauß-Jordan-Algorithmus für allgemeine $m \times n$-Matrizen in einer Programmiersprache eigener Wahl und verifiziere damit die gefundene Lösung zu Teil (a).

Übung 8.4: Man löse die lineare Optimierungsaufgabe

$$\begin{aligned} Q(x) &:= x_1 + 3x_2 + x_3 \to \max! \\ x_i &\geq 0, \quad i = 1, \dots, 5, \\ & 5x_1 + 3x_2 \leq 3, \\ & x_1 + 2x_2 + 4x_3 \leq 4, \end{aligned}$$

mit Hilfe des Simplex-Verfahrens. (Hinweis: Man überführe das System durch Einführung von „Schlupfvarablen“ in Normalform und rate eine Startecke.)

Übung 8.5: Man löse die lineare Optimierungsaufgabe

$$\begin{aligned} Q(x) &:= 4x_1 + x_2 + x_3 \to \min! \\ x_i &\geq 0, \quad i = 1, \dots, 3, \\ & 2x_1 + x_2 + 2x_3 = 4, \\ & 3x_1 + 3x_2 + x_3 = 3, \end{aligned}$$

mit Hilfe des Simplex-Verfahrens. Eine Startecke verschaffe man sich durch die Vorlaufrechnung (Phase I).

Übung 8.6: Man löse das lineare Programm

$$\begin{aligned} Q(x) &:= \tfrac{3}{4}x_1 - 20x_2 + \tfrac{1}{2}x_3 - 6x_4 \to \max!, \\ x_i &\geq 0, \quad i = 1, \dots, 4, \\ & \tfrac{1}{4}x_1 - 8x_2 - x_3 + 9x_4 \leq 0, \\ & \tfrac{1}{2}x_1 - 12x_2 - \tfrac{1}{3}x_3 + 3x_4 \leq 0, \\ & x_3 \leq 1, \end{aligned}$$

mit Hilfe des Simplex-Verfahrens unter Verwendung

a) der Auswahlregel (R), ergänzt um die Vorschrift, dass kleinste mögliche $p \in I^0$ zu wählen;

b) der Auswahlregel $(\tilde{R})$.

In beiden Fällen sei $q \notin I^0$ so bestimmt, dass $\gamma_q = \min\{\gamma_k |\, \gamma_k < 0,\ k \notin I^0\}$, und dass q der kleinste dieser Indizes ist.

c) Man implementiere Phase II des Simplex-Algorithmus für lineare Programme in Normalform für den Fall einer bekannten Startecke $x^{(0)} \in M$ und eines Ausgangstableaus. Man verifiziere damit die unter (a) bzw. (b) gefundene Optimallösung.