Philipps-Universität
Marburg
| FB 12
Mathematik und Informatik

Daniel Hohmann & Thorsten Ante
www.mathematik.uni-marburg.de/∼ante
www.mathematik.uni-marburg.de/∼hohmann

Lineare Optimierung

Vorlesung: WS 06/07 - Prof. Dr. Ekaterina Kostina

Inhaltsverzeichnis

1 Einführung

1.1 Überblick

Bei einem Optimierungsproblem versucht man in der Regel eine **Zielfunktion** f auf einer Grundmenge S von Möglichkeiten zu maximieren, also das $\max_{x\in S} f(x)$ zu finden. Gesucht ist demnach ein $x^* \in X$, sodass

$$f(x^*) \geq f(x) \quad \forall\, x \in S\ .$$

Nichtlineare Optimierungsprobleme: Bei einem nichtlinearen Optimierungsproblem hat man zu der Zielfunktion f so genannte **Nebenbedingungen** oder auch **Restriktionen** g_i und h_j, diffbare Funktionen von $\mathbb{R}^n \to \mathbb{R}$ gegeben mit

$$\begin{matrix} g_i(x) = 0,\ i = 1, ..., m \\ h_j(x) \leq 0,\ j = 1, ..., p \end{matrix} \quad , \quad x \in \mathbb{R}^n\ .$$

Diese schränken die Grundmenge S und somit auch den optimalen Punkt x^* ein.

Konvexe Optimierungsprobleme: Eine Menge $S \subset \mathbb{R}^n$ heißt konvex, falls für alle $x, y \in S$ und $\lambda \in [0, 1]$ gilt

$$\lambda x + (1 - \lambda)y \in S\ .$$

Eine Funktion $f : \mathbb{R}^n \to \mathbb{R}$ heißt konvex, falls für alle $x, y \in S$ und $\lambda \in [0, 1]$ gilt

$$\lambda f(x) + (1 - \lambda)f(y) \geq f(\lambda x + (1 - \lambda)y)\ .$$

Von einem konvexen Optimierungsproblem spricht man nun, wenn man eine konvexe Zielfunktion f und eine konvexe Menge S, z.B. $S = \{x \in \mathbb{R}^n | h_j(x) \text{ konvex},\ j = 1, ..., p\}$, mit dem Ziel $\min_{x\in S} f(x)$ gegeben hat.

Lineare Optimierungsprobleme: Bei einem linearen Optimierungsproblem oder linearen Programm geht man wie der Name schon sagt von einer linearen Zielfunktion und auch linearen Restriktionen aus. Diese sind in der Regel durch zwei Vektoren $c \in \mathbb{R}^n$, $b \in \mathbb{R}^m$ und einer Matrix $A \in \mathbb{R}^{m\times n}$ gegeben. Damit kann man ein solches lineares Programm definieren durch

$$\max_{x\in\mathbb{R}^n} c^T x = \max_{x\in\mathbb{R}^n} \sum_{j=1}^{n} c_j x_j\ ,$$

$$Ax \leq b \quad \Leftrightarrow \quad \begin{matrix} \sum_{j=1}^n a_{1j} x_j \leq b_1 \\ \vdots \\ \sum_{j=1}^n a_{mj} x_j \leq b_m \end{matrix}\ .$$

Lineare gemischt-ganzzahlige Optimierungsprobleme: Bei einem solchen Optimierungsproblem besteht zu einem gewöhnlichen linearen Optimierungsproblem der einzige Unterschied darin, dass ein Teil der Variablen nur aus ganzen Zahlen bestehen darf. Ein Problem ist also durch

$$\max_{x,y} c^T x + d^T y \ ,$$

$$A\begin{pmatrix} x \\ y \end{pmatrix} \leq b$$

für $x \in \mathbb{R}^n$, $y \in \mathbb{Z}^p$, $c \in \mathbb{R}^n$, $d \in \mathbb{R}^p$, $b \in \mathbb{R}^m$ und $A \in \mathbb{R}^{m \times n+p}$ gegeben.

1.2 Beispiele

Beispiel 1.1 Bootsverleihproblem: Ein Bootsverleiher hat die zwei Bootstypen „Standard“ und „Luxus“ im Angebot. Die Einnahmen pro Boot und Ausleihe betragen 600 € für den Typ Standard und 800 € für den Typ Luxus. Für den Verleih stellt sich nun die Frage, wie viele Boote man von jedem Typ haben sollte, um seinen Gewinn zu maximieren.
Hierzu sind jedoch folgende Restriktionen zu berücksichtigen:

(1) maximal 350 Boote

(2) davon höchstens 200 Boote des Typs Luxus

(3) man sollte mehr Boote vom Typ Luxus als vom Typ Standard anbieten

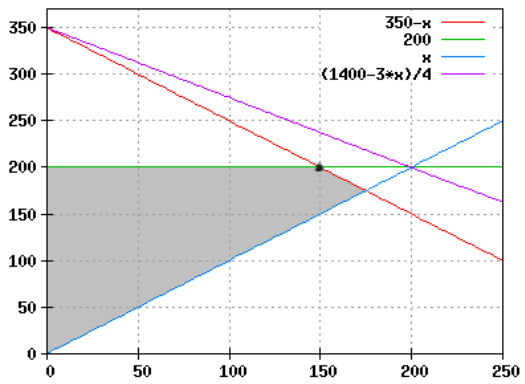
(4) der Wartungsaufwand für alle Boote darf 1400 h/Woche nicht überschreiten, wobei ein Boot vom Typ Standard 3 h/Woche und ein Boot vom Typ Luxus 4h/Woche benötigt

Mathematische Formulierung des Problems: Seien x_1 die Anzahl der Standardboote und x_2 die Anzahl der Luxusboote, so gilt es die Zielfunktion

$$600x_1 + 800x_2 \to \max_x$$

zu maximieren, ohne dabei die Nebenbedingungen

$$\begin{aligned} x_1 + x_2 &\leq 350 \\ x_2 &\leq 200 \\ x_2 &\geq x_1 \\ 3x_1 + 4x_2 &\leq 1400 \\ x_1, x_2 &\geq 0 \end{aligned}$$



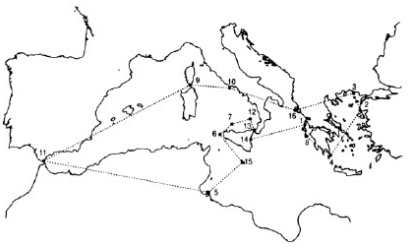
zu verletzen.

Beispiel 1.2 Diät-Problem (Zusammenstellung von Lebensmitteln): Für einen Ernährungsplan stehen n Lebensmittel zu Verfügung. Die jeweiligen Kosten der einzelnen Lebensmittel bezeichne man mit c_i für $i = 1, \ldots, n$. Jedes dieser Lebensmittel enthält natürlich verschiedene wichtige Bestandteile, wie z.B. Vitamin A, Calcium, u.s.w.. In diesem Diätplan beschränkt man sich auf m wichtige Bestandteile und definiert $a_{ji},\ j = 1, ..., m,\ i = 1, ..., n$ als den Anteil des Bestandteils j im Nahrungsmittel i. Die vorgesehene Mindestverpflegung jedes Bestandteils in einer Zeitperiode bezeichne man weiter mit $b_j,\ j = 1, ..., m$.
Nun stellt sich die Frage, welche und wieviel Nahrungsmittel man in einer Zeitperiode kaufen sollte oder muss, um die Kosten für die benötigten Lebensmittel zu minimieren.
Mathematisches Modell: Für $i = 1, ..., n$ sei x_i die Menge des Lebensmittels i.

$$c_1 x_1 + \ldots + c_n x_n = c^T x \to \min_x$$

$$\begin{array}{rcl} a_{11}x_1 + \ldots + a_{1n}x_n & \geq & b_1 \\ \vdots & & \vdots \\ a_{m1}x_1 + \ldots + a_{mn}x_n & \geq & b_m \end{array}$$

Beispiel 1.3 TSP (Travelling Salesman Problem): Gegeben seien n Orte. Mit w_{ij} bezeichne man die Entfernung von Ort i zu Ort j. Gesucht ist nun eine optimale Tour für den Handelsreisenden, also die kürzeste Strecke, die jeden Ort genau einmal besucht und am Ende wieder am Ausgangsort ankommt. Für die mathematische Formulierung des symmetrischen TSP mit $w_{ij} = w_{ji}$ betrachtet man nur die w_{ij} mit $i < j$ und definiert sich dazu passend $x = (x_{ij}) \in \mathbb{B}^k,\ k = \frac{n(n-1)}{2},\ i < j$, mit



$$x_{ij} = \begin{cases} 0\ , & \text{der Weg zwischen } i \text{ und } j \text{ wird nicht benutzt} \\ 1\ , & \text{sonst} \end{cases} .$$

Um nun die gewünschte Route zu erhalten benötigt man zwei Nebenbedingungen:

$$(1) \quad \sum_{j<i} x_{ji} + \sum_{j>i} x_{ij} = 2$$
$$(2) \quad \sum_{i \in U, j \notin U} x_{ij} \geq 2 \quad \forall U \subset \{1, ..., n\},\ 1 \leq |U| \leq n-1$$

Hierbei verhindert Bedingung (1), dass ein Ort mehr als 2mal bereist wird. Durch diese Restriktion kann zu jedem Ort nur ein Weg hin und ein Weg weg führen. Bedingung (2) verhindert, dass in einer echten Teilmenge der Orte ein Kreis auftritt, sprich sich mehrere Ortsgruppen bilden in denen zwar jeder Ort genau einmal bereist wird, es aber keine Verbindung zwischen den einzelnen Gruppen gibt.

1.3 Standardformen für Lineare Programme

Ein lineares Programm (LP) ist in der allgemeinen Form immer gegeben durch eine lineare Ziefunktion

$$c^T x = c_1 x_1 + \ldots + c_n x_n$$

und lineare Ungleichung und Gleichungen der Form

$$\begin{array}{ll} a_{i1}x_1 + \ldots + a_{in}x_n \leq b_i \ , & i = 1, \ldots, m_1 \\ a_{i1}x_1 + \ldots + a_{in}x_n = b_i \ , & i = m_1 + 1, \ldots, m \end{array} .$$

In der Regel betrachtet man hierbei nur nichtnegative Vektoren x. Für den allgemeinen Fall muss man jedoch auch freie Variablen zulassen und definiert deshalb

$$\begin{array}{ll} x_j \geq 0 \ , & j = 1, \ldots, n_1 \\ x_j \text{ frei} \ , & j = n_1 + 1, \ldots, n \end{array} .$$

In diese Form lässt sich nun durch verschiedene Umformungen jedes lineare Programm überführen. Hat man denn z.B. die Restriktion $x_k \leq 0$ vorgegeben, so definiert man sich $\overline{x_k} := -x_k \geq 0$, oder benötigt man $\sum_{j=1}^{n} a_{ij}x_j \geq b_i$, so definiert man sich $\overline{a_{ij}} := -a_{ij},\ j = 1, ..., n$, und $\overline{b_i} := -b_i$. Auch die Vorgabe $\max_x c^T x$ lässt sich einfach anpassen durch $\min_x \overline{c^T} x$ mit $\overline{c_i} := -c_i$.

1.3.1 Kanonische Formen

Seien $x \in \mathbb{R}^n,\ x \geq 0,\ c \in \mathbb{R}^n,\ b \in \mathbb{R}^m,\ A \in \mathbb{R}^{m \times n}$, so ist

1. Standardform	$\min c^T x$ unter $Ax = b$
2. Ungleichungsform	$\min c^T x$ unter $Ax \leq b$

Auf eine dieser beiden Formen lässt sich nun jedes lineare Programm bringen. Außerdem sind die beiden kanonischen Formen äquivalent: Will man ein Problem in der Standardform in die Ungleichungsform überführen, so definiert man sich einfach

$$\tilde{A} := \begin{pmatrix} A \\ -A \end{pmatrix} , \quad \tilde{b} := \begin{pmatrix} b \\ -b \end{pmatrix} .$$

Somit hat man jede Gleichung der Standardform durch zwei Ungleichungen ausgedrückt. Will man ein Problem in der Ungleichungsform in die Standardform überführen, so ist es nötig so genannte *Schlupfvariablen* z_j mit $0 \leq z \in \mathbb{R}^m$ einzuführen. Durch

$$\tilde{A} := (A, I) , \quad \tilde{c}^T := (c, 0) , \quad \tilde{x} := \begin{pmatrix} x \\ z \end{pmatrix}$$

hat man somit alle Ungleichungen der Ungleichungsform durch Gleichungen ausgedrückt. Außerdem kann man durch

$$x_j := x_j^+ - x_j^- \quad \text{mit } x_j^+, x_j^- \geq 0$$

auch freie Variablen x_j in eine nichtnegative Form überführen.

Notation Um den später auftretenden *Simplex-Algrithmus* übersichtlicher zu gestalten werden folgenden Notationen eingeführt:

- $A = (a_{ij})_{i=1,\dots,m,\ j=1,\dots,n}$
- $A_{\cdot j} := \begin{pmatrix} a_{1j} \\ \vdots \\ a_{mj} \end{pmatrix}$, $A_{i\cdot} := (\, a_{i1}, \dots, a_{in}\,)$
- $I := \{1, \dots, m\}$ Menge der Zeilenindizes
- $J := \{1, \dots, n\}$ Menge der Spaltenindizes
- $I \supset \tilde{I} := \{i_1, \dots, i_p\},\ i_k \in I$ Teilmenge der Zeilenindizes
- $J \supset \tilde{J} := \{j_1, \dots, j_q\},\ j_k \in J$ Teilmenge der Spaltenindizes
- $A_{\tilde{I}\tilde{J}} := (a_{ij})_{i\in\tilde{I},\ j\in\tilde{J}}$

2 Geometrische Interpretation

2.1 Grundlagen der konvexen Geometrie

Definition 2.1 *Eine Menge $C \subset \mathbb{R}^n$ heißt* **konvex**, *wenn*

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

Definition 2.2

(a) x heißt eine **konvexe Kombination** *von $x_1, \ldots, x_N$, wenn ein $0 \leq \lambda \in \mathbb{R}^N$ mit $\sum_{i=1}^N \lambda_i = 1$ existiert, sodass für $A := (x_1, \ldots, x_N)$ gilt*

$$\sum_{i=1}^{N} \lambda_i x_i = A\lambda = x\ .$$

(b) $Conv(A) := \{x | x$ konv. Kombination von $A\}$ heißt **konvexe Hülle** *von A.*

Definition 2.3

(a) $C \subset \mathbb{R}^n$ heißt ein **Kegel**, *wenn*

$$\forall\ x \in C,\ \lambda \geq 0: \quad \lambda x \in C\ .$$

(b) $\{x = \sum_{k=1}^N \alpha_k x_k | \alpha_k \geq 0\}$ heißt der von $x_1, \ldots, x_N$ erzeugte **konvexe Kegel**.

Definition 2.4 *x heißt* **Extremalpunkt** *einer konvexen Menge C, wenn x keine konvexe Linearkombination von zwei (oder mehreren Punkten) aus C ist.*

Definition *Sei $a \in \mathbb{R}^n,\ a \neq 0,\ \beta \in \mathbb{R}$, so heißt*

(i) $\bar{H} := \{x | a^T x \leq \beta\}$ ein **Halbraum**,

(ii) $H := \{x | a^T x = \beta\}$ eine **Hyperebene**

(iii) und a der **Normalenvektor** *zu H bzw. $\bar{H}$.*

Bemerkung

(i) a steht orthogonal auf H, d.h. für alle $x, y \in H$ gilt $a^T(x-y) = 0$.

(ii) a ist die Richtung, in die $f(x) := a^T x$ wächst, denn für $x \in H$ und $\varepsilon > 0$ gilt

$$a^T(x + \varepsilon a) = a^T x + \varepsilon \underbrace{a^T a}_{>0} > a^T x \ .$$

Definition 2.6 *$S \subset \mathbb{R}^n$ heißt* **affiner Raum**, *wenn*

$$\forall\ x_1, x_2 \in S,\ \lambda \in \mathbb{R}: \quad \lambda x_1 + (1-\lambda) x_2 \in S \ ,$$

oder allgemeiner wenn

$$\forall\ x_1, ..., x_N \in S,\ \lambda_1, ..., \lambda_N \in \mathbb{R} \text{ mit } \sum_{i=1}^{N} \lambda_i = 1: \quad \sum_{i=1}^{N} \lambda_i x_i \in S \ .$$

Bemerkung Im Gegensatz zu einer konvexen Menge muss bei einem affinen Raum nicht nur die Strecke zwischen zwei Punkten, sondern die gesamte Gerade durch diese zwei Punkte wieder in der Menge liegen.

Definition 2.7

(a) Eine Menge $P \subset \mathbb{R}^n$ heißt **Polyeder**, *wenn P der Durschnitt endlich vieler Halbräume (und Hyperebenen) ist, also*

$$\begin{aligned} P &= \bigcap_{i=1}^{m} \{x | a_i^T x \leq b_i\} \ , \quad a_i \in \mathbb{R}^n,\ b_i \in \mathbb{R} \\ &= \{x | Ax \leq b\} \ , \quad A = \begin{pmatrix} a_1^T \\ \vdots \\ a_m^T \end{pmatrix},\ b = \begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix} \ . \end{aligned}$$

(b) Ein Polyeder $P \neq \varnothing$ heißt **Polytop**, *wenn er beschränkt ist, d.h.*

$$\exists\ k \in \mathbb{R},\ k > 0: \quad P \subset \{x \in \mathbb{R}^n |\ \|x\| \leq k\} \ .$$

Bemerkung Eine Hyperebene $\{x | a^T x = b\}$ ist der Durchschnitt der zwei Halbräume $\{x | a^T x \leq b\}$ und $\{x | a^T x \geq b\}$.

Lemma Hyperebenen, Halbräume und ihre Durchschnitte (damit auch Polyeder und Polytope) sind konvex.

Korollar Die zulässige Menge eines Linearen Programms in der Standardform $\{x|Ax = b,\ x \geq 0\}$ oder in der Ungleichungsform $\{x|Ax \leq b,\ x \geq 0\}$ ist ein Polyeder und somit konvex.

Definition 2.8 *Die* **Dimension** *eines linearen Unterraums und eines dazu parallelen affinen Raums ist die maximale Anzahl linear unabhängiger Vektoren. Die Dimension einer Teilmenge $D \subset \mathbb{R}^n$ (z.B. ein Polyeder) ist die Dimension des kleinsten affinen Raums, der D enthält.*

Definition 2.9 *Sei C eine konvexe Menge und H die Hyperebene zum Halbraum $\bar{H}$. Ist $C \cap H \neq \varnothing$ und $C \subset \bar{H}$, so heißt H eine* **Stützhyperebene** *an C.*

Korollar Sei $P = \{x|Ax \leq b\}$ ein Polyeder (ohne redundante Ungleichungen), so ist $\forall\ 1 \leq i \leq m$

$$H_i := \{x|A_i.x = b_i\}$$

($A_i.$ ist die i-te Zeile von A, vgl. Kap. Notation) eine Stützhyperebene an P.

Definition 2.10 *Sei P ein Polyeder der Dimension d und H eine Stützebene von P, so bezeichnet man den Durchschnitt $H \cap P$ als* **Seite**. *Diese unterscheidet man nun wieder in Abhängigkeit von ihrer Dimension:*

Dimension	*Bezeichnung*
0	**Ecke**
1	**Kante**
$d-1$	**Facette**

Bemerkung:

(i) Jede Facette des Polyeders $P = \{x|Ax \leq b,\ x \geq 0\}$ korrespondiert zu einem Halbraum oder einer Hyperebene.

(ii) Ecken sind Extremalpunkte.

2.2 Extremalpunkte und Basislösungen

Sei im Folgenden $P = \{x \in \mathbb{R}^n | Ax = b,\ x \geq 0\}$, $\quad A \in \mathbb{R}^{m \times n},\ m \leq n$.

Definition *Ist $x \in P$, so heißt $I(x) := \{i | 1 \leq i \leq n,\ x_i = 0\}$ die Menge der* **aktiven Indizes**.

Satz 2.11 $x \in P$ ist genau dann eine Ecke von P, wenn die Spalten j der Matrix A mit $j \notin I(x)$ linear unabhängig sind.

Beweis Es sei o.E. $x_1, \ldots, x_p > 0$ und $x_{p+1} = \ldots = x_n = 0$, also $I(x) = \{p+1, ..., n\}$. Definiere nun

$$x = \begin{pmatrix} \bar{x} \\ 0 \end{pmatrix} \quad \text{und passend } A = (\bar{A}, \tilde{A}) , \quad A \in \mathbb{R}^{m \times p + (n-p)} .$$

Somit ist also $\quad Ax = b \ \Leftrightarrow\ \bar{A}\bar{x} + \tilde{A}0 = \bar{A}\bar{x} = b$.

„⇒“ Sei x eine Ecke. Annahme: $\bar{A}$ hat nicht vollen Rang. Somit existiert ein $w \neq 0$ mit $\bar{A}w = 0$ und außerdem ein $\varepsilon > 0$, sodass $\bar{x} \pm \varepsilon w > 0$. Demnach ist also auch

$$\bar{A}(\bar{x} \pm \varepsilon w) = \bar{A}\bar{x} \pm \varepsilon \bar{A}w = \bar{A}\bar{x} = b .$$

Die Punkte y_1 und y_2 mit

$$y_1 := \begin{pmatrix} \bar{x} - \varepsilon w \\ 0 \end{pmatrix} \geq 0 , \quad y_2 := \begin{pmatrix} \bar{x} + \varepsilon w \\ 0 \end{pmatrix} \geq 0$$

sind demnach auch zulässige Punkte, also in P, aber da nun $x = \frac{1}{2}(y_1 + y_2)$, kann x keine Ecke sein. Da dies ein Widerspruch zu der Voraussetzung ist, muss $\bar{A}$ also vollen Rang haben.

„⇐“ Habe nun $\bar{A}$ vollen Rang. Annahme: x ist keine Ecke. Somit existieren $x \neq x', x'' \in P$ mit $x = \lambda x' + (1 - \lambda)x''$ für $\lambda \in]0, 1[$. Demnach sind auch $Ax' = Ax'' = b$, also $A(x - x') = 0$. Da nun aber für jedes $i \in I(x)$ gilt

$$0 = x_i = \underbrace{\lambda}_{>0} \underbrace{x_i'}_{\geq 0} + \underbrace{(1-\lambda)}_{>0} \underbrace{x_i''}_{\geq 0} ,$$

folgt $x_i' = x_i'' = 0\ \forall\ i \in I(x)$. Demnach ist jedoch wieder schon $\bar{A}\bar{x'} = b$, also

$$\bar{A}(\bar{x} - \bar{x'}) = 0 .$$

Da aber aus $x \neq x'$ und $x_i = x_i' = 0\ \forall\ i \in I(x)$ folgt $\bar{x} \neq \bar{x'}$, hat man mit $\bar{x} - \bar{x'}$ einen Vektor $x^* \neq 0$ gefunden mit $\bar{A}x^* = 0$. Somit kann aber $\bar{A}$ nicht vollen Rang haben. Da dies aber wieder ein Widerspruch zu den Voraussetzungen darstellt, muss x also eine Ecke sein. ■

Bemerkung Falls gilt $Rang(A) = m$, so kann man die Ecken von P einfach und eindeutig beschreiben. Ist jedoch $Rang(A) \neq m$, so hat $Ax = b$ entweder keine Lösungen, d.h. $P = \varnothing$, oder $Ax = b$ besitzt redundante Zeilen, welche eliminiert werden können.

Im Folgenden gelte: *Rang* $A = m \leq n$

Definition 2.12 *Sei* Π *eine beliebige Permutation in* $\mathbb{R}^n$*, sodass*

$$A \cdot \Pi = (\bar{A}, \tilde{A})$$

mit $\bar{A} \in \mathbb{R}^{m \times m}$*,* $\tilde{A} \in \mathbb{R}^{m \times (n-m)}$ *und* $\bar{A}$ *regulär.*

(i) Setze alle Komponenten von x *zu* $\tilde{A}$ *auf* 0*, also* $\tilde{x} = 0$ *die* **Nichtbasisvariablen**.

(ii) Die Komponenten von x *zu* $\bar{A}$ *heißen* **Basisvariablen**. *Nun gilt*

$$Ax = b \quad \Leftrightarrow \quad (\bar{A}, \tilde{A}) \begin{pmatrix} \bar{x} \\ \tilde{x} \end{pmatrix} = b \quad \Leftrightarrow \quad \bar{A}\bar{x} = b \quad \Leftrightarrow \quad \bar{x} = \bar{A}^{-1} b \ .$$

(iii) $x = \begin{pmatrix} \bar{A}^{-1} b \\ 0 \end{pmatrix}$ *heißt* **Basislösung**.

(iv) Ist $\bar{A}^{-1} b \geq 0$ *(d.h.* $0 \leq x \in P$ *zulässig) so nennt man* x *eine zulässige Basislösung. Dies ist genau dann der Fall, wenn* x *eine Ecke von* P *ist.*

Korollar 2.13 $x \in P$ ist genau dann eine Ecke von P, wenn eine Matrix $\bar{A}$ existiert, sodass x eine zulässige Basislösung ist.

Korollar 2.14 P hat nur endlich viele Ecken.

Beweis Es existieren nur endlich viele Permutationen, und zwar höchstens $\binom{n}{m} = \frac{n!}{m!(n-m)!}$.

Definition 2.15 *Eine* **Richtung** *ist ein Vektor* $d \in \mathbb{R}^n_+$ *mit*

$$x_0 + \lambda d \in P \qquad \forall \ x_0 \in P, \ \lambda \geq 0 \ .$$

Offenbar gilt:

(i) P *ist unbeschränkt* $\quad \Leftrightarrow \quad \exists$ *eine Richtung*

(ii) Da $x_0 + \lambda d \in P$ *ist also* $Ad = 0$.

Satz 2.16 (Darstellungsatz) Jeder Punkt $x \in P$ hat eine Darstellung

$$x = \sum_{i \in V} \lambda_i v_i + d \qquad \text{mit} \ \sum_{i \in V} \lambda_i = 1, \ \lambda_i \geq 0 \ \forall i \in V \ ,$$

wobei $\{v_i | i \in V\}$ die Menge der Ecken von P und d eine Richtung oder 0 ist. Jeder Punkt in P lässt sich somit als (affine) Linearkombination der Ecken schreiben.

Beweis per Induktion über p (Anzahl der positiven Komponenten von x)
Ind.-Anfang: $p = 0 \quad \Rightarrow \quad x = 0 \quad \Rightarrow \quad x$ ist eine Ecke von P

Annahme: Die Aussage gilt für $p - 1 > 0$

Ind.-Schritt: Habe x nun p positive Komponenten

a) x ist eine Ecke, so ist $x = v_i$ für ein $i \in V$

b) x ist keine Ecke, dann konstruiere $w \neq 0$ mit $w = \binom{\bar{w}}{0}$, $\bar{w} \neq 0$, $w_i = 0$ zu $x_i = 0$ (n-p Komponenten)

1.Fall: w hat sowohl positive, als auch negative Komponenten. Betrachte nun

$$x(\theta) = x + \theta w = \begin{pmatrix} \bar{x} + \theta \bar{w} \\ 0 \end{pmatrix} .$$

Für welche θ ist nun $x(\theta) \geq 0$? Für

$$\theta_i = \begin{cases} -\frac{\bar{x}_i}{\bar{w}_i} & , \text{ falls } \bar{w}_i < 0 \\ \infty & , \text{ falls } \bar{w}_i \geq 0 \end{cases} .$$

Berechne nun $\theta' > 0$ mit $\theta' := \min \theta_i^+$, wobei $\theta_i^+ = \begin{cases} -\frac{\bar{x}_i}{\bar{w}_i} & , \text{ falls } \bar{w}_i < 0 \\ \infty & , \text{ sonst} \end{cases}$ und $\theta'' < 0$ mit $|\theta''| := \min |\theta_i^-|$, wobei $\theta_i^- = \begin{cases} -\frac{\bar{x}_i}{\bar{w}_i} & , \text{ falls } \bar{w}_i > 0 \\ \infty & , \text{ sonst} \end{cases}$.
Nun kann man $\underbrace{x(\theta')}_{x'}$ und $\underbrace{x(\theta'')}_{x''}$ berechnen, welche beide haben höchstens $p-1$ positive Komponenten besitzen, da $x(\theta') = x + \theta' w = \binom{\bar{x}+\theta' w}{0}$ und mind. ein j ex. mit $\bar{x}_j + \theta' w_j = 0$. Somit ist nun

$$x = \mu x' + (1-\mu)x'' = \mu \left(\sum_{i \in V} \lambda_i' v_i + d' \right) + (1-\mu) \left(\sum_{i \in V} \lambda_i'' v_i + d'' \right)$$

$$= \sum_{i \in V} (\mu \lambda_i' + (1-\mu)\lambda'') v_i + (\mu d' + (1-\mu) d'') , \quad \text{wobei } \mu = -\frac{\theta''}{\theta' - \theta''}$$

$$\Rightarrow \quad x = \mu(x + \theta' w) + (1-\mu)(x + \theta'' w) = x + \underbrace{(\mu\theta' + (1-\mu)\theta'')}_{=0} w$$

Nun ist $0 < \mu < 1$, da $\mu = -\frac{-\theta'' + \theta' - \theta'}{\theta' - \theta''} = 1 - \frac{\theta'}{\theta' - \theta''}$. Außerdem sind $\lambda_i', \lambda_i'' \geq 0$ und $\sum_{i \in V} \lambda_i' = \sum_{i \in V} \lambda_i'' = 1 \ \forall \ i \in V$. Weiter sind $d', d'' \geq 0$ mit $Ad' = Ad'' = 0$, also Richtungen. Deshalb definiert man nun

$$\lambda_i := \mu \lambda_i' + (1-\mu)\lambda_i'' \geq 0 \ , \ i \in V, \text{ mit } \sum_{i \in V} \lambda_i = 1 \ ,$$

$$d := \mu d' + (1-\mu) d'' \geq 0 \ , \ Ad = 0 \qquad \text{(also Richtung)}$$

und somit ist

$$x = \sum_{i \in V} \lambda_i v_i + d \ .$$

2.Fall: Sei $\bar{w} < 0$. Nun ist nur die Berechnung von θ' und $x' = x(\theta')$ nötig und es ergibt sich

$$x = x(\theta') - \theta' w = \sum_{i \in V} \lambda_i' v_i + \underbrace{d' - \theta' w}_{=:d} \ .$$

3. Fall: $\bar{w} > 0$, so berechnet man $\theta'' < 0$ und $x'' = x(\theta'')$.

■

Korollar 2.17 Ein Polytop (d.h. beschränkt und nicht leer) hat die Darstellung

$$x \in P \quad \Leftrightarrow \quad x = \sum_{i \in V} \lambda_i v_i \ ,$$

mit $\sum_{i \in V} \lambda_i = 1 \ , \ \lambda_i \geq 0$.

Satz 2.18 (Fundamentalsatz der LP) Sei P nicht leer, so

(i) existiert mindestens eine Ecke in P und

(ii) $c^T x$ hat entweder kein Minimum in P (d.h. $c^T x \to \infty$) oder das Minimum wird in einer Ecke angenommen.

Beweis (i) Folgt direkt aus dem Darstellungssatz

(ii) 1.Fall: Annahme: Es existiert eine Richtung d, sodass $c^T d < 0$, so gilt für alle x:

$$x = \underbrace{\sum_{i \in V} \lambda_i v_i}_{=:\tilde{x}} + d \ .$$

Somit kann $c^T x = c^T \tilde{x} + \underbrace{c^T d}_{<0}$ beliebig klein werden.

2.Fall: Annahme: Es existiert keine Richtung d mit $c^T d < 0$, also $\exists \ d$ mit $c^T d \geq 0$. Dann ist

$$c^T \tilde{x} = \sum_{i \in V} \lambda_i c^T v_i \geq \sum_{i \in V} \lambda_i (\min_{i \in V} c^T v_i) = \min_{i \in V} c^T v_i \ .$$

Somit gilt $\forall \ \tilde{x} \in P$: $c^T \tilde{x} \geq \min_{i \in V} c^T v_i$. Das Minimum wird somit in einer Ecke angenommen. ■

3 Die Simplex-Methode

Den Durchbruch für die lineare Optimierung schaffte George Dantzig im Jahre 1947, als er eine Arbeit über das Simplex-Verfahren veröffentlichte, welches heute eines der meistgenutzten Verfahren zur Lösung linearer Programme ist.

Die Idee des Simplex-Verfahrens:

- Starte in einer Ecke x^{alt}
- Optimal? Falls ja, dann STOP
- Nein $\Rightarrow$ $\exists$ eine Nachbarecke x^{neu} mit besseren Werten, also $c^T x^{neu} \leq c^T x^{alt}$

3.1 Basen und ihre zugehörigen Basislösungen

Im Folgenden gelte wieder $A \in \mathbb{R}^{m \times n}$ mit $m < n$ und *Rang* $A = m$. Außerdem sei $P = \{x | Ax = b,\ x \geq 0\} \neq \varnothing$.

Definition 3.1 *Die Indexmenge $B := \{j_1, \ldots, j_m\} \subset \{1, \ldots, n\}$ heißt* **Basis**, *falls die Matrix*

$$A_B := (A_{\cdot j_1}, \ldots, A_{\cdot j_m})$$

invertierbar ist, also vollen Rang besitzt. A_B heißt **Basismatrix**. *Die Indexmenge $N := \{1, \ldots, n\} \backslash B$ heißt die Menge der inaktiven oder nichtbasis Indizes.*

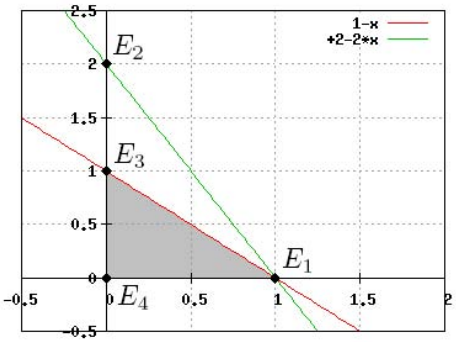
Definition 3.1a *Setzt man x_N=0, so heißt x* **Basislösung** *zu B. Gilt dann $x_B \geq 0$, so ist $x \in P$ und somit eine zulässige Basislösung.*

Definition 3.2 *Eine Ecke heißt* **degeneriert**, *falls ein oder mehrere $j \in B$ existieren mit $x_j = 0$.*

Beispiel (Degeneriertheit) Sei das folgende LGS gegeben:

$$\begin{array}{rcl} x_1 + x_2 & \leq & 1 \\ 2x_1 + x_2 & \leq & 2 \\ x_1, x_2 & \geq & 0 \end{array} \quad \Leftrightarrow \quad \begin{array}{rcl} x_1 + x_2 + z_1 & = & 1 \\ 2x_1 + x_2 + z_2 & = & 2 \\ x_1, x_2 & \geq & 0 \end{array}$$

Somit also $A = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 2 & 1 & 0 & 1 \end{pmatrix}$ und $x^T = (x_1, x_2, z_1, z_2)$.



In der folgenden Tabelle bestimmt man alle Basen und die zugehörigen Basislösungen:

B	A_B	N	x_N	x_B	x	Ecke
$\{1,2\}$	$\begin{pmatrix} 1 & 1 \\ 2 & 1 \end{pmatrix}$	$\{3,4\}$	$(z_1, z_2) = (0,0)$	$(x_1, x_2) = (1,0)$	$(1,0,0,0)$	E_1
$\{1,3\}$	$\begin{pmatrix} 1 & 1 \\ 2 & 0 \end{pmatrix}$	$\{3,4\}$	$(x_2, z_2) = (0,0)$	$(x_1, z_1) = (1,0)$	$(1,0,0,0)$	E_1
$\{1,4\}$	$\begin{pmatrix} 1 & 0 \\ 2 & 1 \end{pmatrix}$	$\{2,3\}$	$(x_2, z_1) = (0,0)$	$(x_1, z_2) = (1,0)$	$(1,0,0,0)$	E_1
$\{2,3\}$	$\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$	$\{1,4\}$	$(x_1, z_2) = (0,0)$	$(x_2, z_1) = (2,-1)$	$(0,2,-1,0)$	E_2
$\{2,4\}$	$\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$	$\{1,3\}$	$(x_1, z_1) = (0,0)$	$(x_2, z_2) = (1,1)$	$(0,1,0,1)$	E_3
$\{3,4\}$	$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$	$\{1,2\}$	$(x_1, x_2) = (0,0)$	$(z_1, z_2) = (1,2)$	$(0,0,1,2)$	E_4

Bemerkung Ist x eine nicht degenerierte Ecke, so existiert genau eine Basis zu x (bis auf die Reihenfolge innerhalb von B und N). Ist x degeneriert so können, wie man an dem obigen Beispiel sieht, mehrere Basen zu x existieren. E_1 ist also eine solche degenerierte Ecke. Außerdem ist die Ecke E_2 keine zulässige Basislösung, da sie eine negative Komponente besitzt.

3.2 Analyse der Zielfunktion

Wie in der „Idee des Simplex-Verfahrens“ beschrieben, startet man mit dem Verfahren also in einer dieser Ecken. Nun stellt sich aber direkt die Frage, wie man herausfindet ob diese Ecke schon optimal ist, und wenn nicht wie man dann die nächstgelegenen Ecken in dem Polyeder P findet. Habe man dafür nun die Indexmengen B und N gegeben, so ist $A = (A_B, A_N)$, $x = \binom{x_B}{x_N}$, $c^T = (c_B, c_N)$ und

$$\begin{array}{ccc} \min c^T x & & \min c_B^T x_B + c_N^T x_N \\ Ax = b & \Leftrightarrow & A_B x_B + A_N x_N = b \\ x \geq 0 & & x_B, x_N \geq 0 \end{array} .$$

Ist nun x eine Ecke, so erfüllt diese das Gleichungssystem

$$\underbrace{\begin{pmatrix} A_B & A_N \\ 0 & I \end{pmatrix}}_{=:M} \begin{pmatrix} x_B \\ x_N \end{pmatrix} = \begin{pmatrix} b \\ 0 \end{pmatrix} .$$

$M \in \mathbb{R}^{n \times n}$ ist invertierbar. Die Kanten, die von der Ecke x ausgehen, erhält man nun durch das Lösen den LGS

$$M \begin{pmatrix} \eta_B^i \\ \eta_N^i \end{pmatrix} = \begin{pmatrix} 0 \\ e_i \end{pmatrix} \quad \text{für } i = 1, \ldots, n-m ,$$

oder genauer

$$\begin{pmatrix} \eta_B^i \\ \eta_N^i \end{pmatrix} = \underbrace{\begin{pmatrix} A_B^{-1} & -A_B^{-1} A_N \\ 0 & I \end{pmatrix}}_{=M^{-1}} \begin{pmatrix} 0 \\ e_i \end{pmatrix} = \begin{pmatrix} -A_B^{-1} A_{\cdot j_{m+i}} \\ e_i \end{pmatrix} .$$

Lemma Der Vektor η^i entspricht genau der $(m+i)$-ten Spalte von M^{-1}. Ist x nicht degeneriert, so sind alle Kantenrichtungen $\eta^i,\ i = 1, \ldots, n-m$, zulässige Richtungen, das heißt

$$x(\theta) = x + \theta\eta^i \in P\ , \quad \forall\ 0 \leq \theta \leq \bar{\theta}\ .$$

Beweis $x(\theta)$ löst das LGS, da $Ax(\theta) = Ax + \theta A\eta^i = b + \theta \underbrace{(A_B\eta^i_B + A_N\eta^i_N)}_{=0 \text{ nach Konstr.}}$. Außerdem ist $x_N(\theta) = x_N + \theta\eta^i_N = \theta e_i \geq 0$ und für hinreichend kleine θ ist auch $x_B(\theta) = \underbrace{x_B}_{>0} + \theta\eta^i_B \geq 0$. ■

Gesucht wird nun die Abstiegsrichtung, d.h. die Kante, in dessen Richtung die Zielfunktion fällt. Hierfür muss die Zielfunktion analysiert werden. Man betrachtet deshalb

$$\begin{aligned} c^T x(\theta) &= c^T x + \theta c^T\eta^i = c^T_B x_B + c^T_N x_N + \theta(c^T_B\eta^i_B + c^T_N\eta^i_N) \\ &= c^T_B x_B + \theta \underbrace{(-c^T_B A^{-1}_B A_{\cdot j_{m+i}} + c_{j_{m+i}})}_{=:\mu_{j_{m+i}}} \\ &= c^T_B x_B + \theta\mu_{j_{m+i}}\ . \end{aligned}$$

Ist also $\mu_{j_{m+i}} < 0$, so ist $c^T x(\theta) \leq c^T x$.

Definition 3.4 *Die Einträge* $\mu_k = c_k - c^T_B A^{-1}_B A_{\bullet k},\ k = 1, \ldots, n,$ *heißen* **reduzierte Kosten**. Für alle $j \in B$ gilt offensichtlich $\mu_j = 0$.

Korollar 3.5 Für alle $i = 1, ..., n-m$ ist $\mu_{k_{m+i}}$ eine Richtungsableitung von $c^T x$ in Richtung η^i. Existiert ein $\mu_{j_{m+1}} < 0$, so ist η^i eine zulässige Abstiegsrichtung von x aus, in der die Zielfunktion fällt.

Bemerkung zur Berechnung von μ

1. Man berechnet zuerst $\pi^T := c^T_B A^{-1}_B$. Man nennt π den Multiplikator (oder Lagrange Variable).
2. Danach kann man vereinfacht alle $\mu_j = c_j - \pi^T A_{\bullet j}$ für $j \in N$ berechnen.

Lemma 3.6 Sei x eine zulässige Basislösung mit B, N (auch degeneriert), so gilt für alle $y \in P$:

$$y = x + \sum_{k=1}^{n-m} \lambda_k\eta^k\ , \quad \text{wobei } \lambda_k \geq 0,\ \eta^k = \begin{pmatrix} -A^{-1}_B A_{\bullet j_{m+k}} \\ e_k \end{pmatrix}$$

Es liegt also jeder Punkt aus P in dem konvexen Kegel von x, der durch die Kanten η^k aufgespannt wird.

Beweis Sei $y \in P$ beliebig, also $y = \binom{y_B}{y_N} \geq 0$ mit $A_B y_B + A_N y_N = b$. Außerdem ist ja $A_B x_B +$

$A_N \underbrace{x_N}_{=0} = A_B x_B = b$ mit $x_B \geq 0$, $x_N = 0$. Demnach ist also

$$A(y-x) = A_B(y_B - x_B) + A_N y_N = 0 \ .$$

$$\Rightarrow y - x = \begin{pmatrix} y_B - x_B \\ y_N - x_N \end{pmatrix} = \begin{pmatrix} -A_B^{-1} A_N y_N \\ y_N \end{pmatrix} = \begin{pmatrix} -A_B^{-1} A_N \\ I \end{pmatrix} y_N = \sum_{k=1}^{n-m} \eta^k y_{j_{m+k}} \quad \text{mit } y_{j_{m+k}} \geq 0$$

■

Korollar 3.7 Ist $\mu_{j_{m+k}} \geq 0$ für alle $1 \leq k \leq n-m$, so folgt aus

$$c^T(y-x) = \sum_{k=1}^{n-m} y_{j_{m+k}} c^T \eta^k = \sum_{k=1}^{n-m} y_{j_{m+k}} \mu_{j_{m+k}} \ ,$$

dass $c^T x \leq c^T y \quad \forall\, y \in P$. x ist in diesem Fall also ein Optimum.

Satz 3.8 Sei x eine Basislösung mit B, N. Falls alle reduzierten Kosten nicht negativ sind, ist x eine optimale Lösung. Dies gilt übrigens auch für degenerierte Ecken.

Korollar 3.9 Sei x nicht degeneriert. Ist $\mu_{j_{m+k}} > 0 \quad \forall\, k = 1, ..., n-m$, so ist x ein eindeutiges Minimum.

Korollar 3.10 Ist x zulässig und optimal und außerdem $\mu_j = 0 \quad \forall j \in K \subseteq N$, so ist auch jeder Punkt $y \in P$ mit $y = x + \sum_{i \in K} \lambda_i \eta^i$ optimal.

3.3 Basiswechsel

Hat man nun eine Basislösung mit negativen Einträgen in ihrem Kostenvektor, so ist der Wert dieser Basislösung in der Zielfunktion also nicht optimal. Die Idee des Simplex-Algorithmus ist es nun, auf einer Kante in Richtung der Zielfunktion zu einer nächsten Basislösung zu „wandern“, um zu schauen, ob deren Wert optimal ist. Nun stellt sich jedoch die Frage, wie man eine solche benachbarte Basislösung findet.

Notation $w = -\eta_B^k$ ist die Korrektur für x_B, d.h. $x_B(\theta) = x_B - \theta w$.

Satz 3.11 Falls $\mu_{j_{m+k}} < 0$ und $w \leq 0$, dann ist η^k eine Richtung in P und das LP hat keine Lösung, weil $c^T \eta^k = \mu_{j_{m+k}} < 0$.

Beweis $c^T\eta^k = (c_B^T, c_N^T)\binom{-A_B^{-1}A_{\bullet j_{m+k}}}{e_k} = -c_B^T A_B^{-1} A_{\bullet j_{m+k}} + c_{j_{m+k}} = \mu_{j_{m+k}} < 0$. Aber $x(\theta) = x - \theta w$ ist zulässig für alle $\theta \geq 0$. ■

Satz 3.12 Ist $\mu_{j_{m+k}} < 0$ und ex. eine positive Komponente in w, so ist

a) $\bar{x} = x + \theta\eta^k$ eine neue Basislösung, wobei

$$\theta := \min_s \left\{ \frac{x_{j_s}}{w_s},\ w_s > 0,\ s = 1, \ldots, m \right\} = \frac{x_{j_q}}{w_q} \geq 0$$

b) und $c^T\bar{x} = c^T x - \theta\mu_{j_{m+k}}$. Ist x nicht degeneriert, also $\theta > 0$, so ist sogar $c^T\bar{x} < c^T x$.

Beweis $x(\theta) = x + \theta\eta^k = \binom{x_B + \theta\eta_B^k}{x_N + \theta e_k} = \binom{x_B - \theta w}{\theta e_k} \overset{!}{\in} P$. Zu zeigen ist also: (i) $Ax(\theta) = b$, (ii) $x(\theta) \geq 0$

(i) $Ax(\theta) = Ax + \theta A\binom{\eta_B^k}{e_k} = b + \theta(A_B\eta_B^k + A_N e_k) = b + \theta(A_B(-A_B^{-1}A_{\bullet j_{m+k}}) + A_{\bullet j_{m+k}}) = b$

(ii) $x_N(\theta) = \theta e_K = (0, ..., \theta, ..., 0)^T \geq 0$ und $x_B(\theta) = x_B - \theta w \overset{!}{\geq} 0$, d.h. $x_{j_s} - \theta w_s \overset{!}{\geq} 0 \quad \forall\ 1 \leq s \leq m$.
Dies jedoch gegeben, wenn man sich definiert $\theta := \min \begin{cases} \frac{x_{js}}{w_s} & , \text{ falls } w_s > 0 \\ \infty & , \text{ falls } w_s \leq 0 \end{cases}$. ■

Bemerkung (i) Satz 3.12 a) gibt gleichzeitig das Verfahren an, mit dem man eine benachbarte Basislösung berechnet. Hierbei „wandert“ der Index $j_{m+k} \in N$ von N nach B. Die Positivitätsbedingung wird dabei nicht verletzt, da $x_{j_{m+k}} = 0$, also $\overline{x_{j_{m+k}}} = \theta \geq 0$. Der Index $j_q \in B$ „wandert“ analog von B nach N. Durch die spezielle Wahl von θ gilt außerdem $\overline{x_{j_q}} = 0$. Die Matrix $\overline{A_B}$ bekommt somit die Gestalt

$$\overline{A_B} = A_B + (-A_{\bullet j_q} + A_{\bullet j_{m+k}})e_q^T \ .$$

Auch $\overline{A_B}$ ist wieder invertierbar.

(ii) Bei dem oben beschriebenen Verfahren ist es möglich, dass man eine neue Basislösung, aber nicht unbedingt ein neues x gefunden hat. Dies ist dann der Fall, wenn $\theta = 0$ war, was bei einer degenerierten Ecke möglich ist.

3.4 Das revidierte Simplex-Verfahren

Zusammenfassung des Verfahrens:

Sei das lineare Programm $\{\min c^T x,\ Ax = b,\ x \geq 0\}$ mit $x \in \mathbb{R}^n,\ b \in \mathbb{R}^m$ und $A \in \mathbb{R}^{m \times n}$ für $n > m$ gegeben. Sei weiter x eine Basislösung, d.h. $B = \{j_1, \ldots, j_m\}$, $N = \{j_{m+1}, \ldots, j_n\}$ mit $x_B \geq 0$, $x_N = 0$ und A_B regulär. Nun kann man die revidierte Simplex-Methode beginnen.

Jeder Iterationsschritt des Verfahrens beinhaltet nun die folgenden Schritte:

(1) Man berechnet zuerst den Multiplikator $\pi^T = c_B^T A_B^{-1}$ und danach die reduzierten Kosten $\mu_N^T = c_N^T - \pi^T A_N$.

(2) Danach kann man x auf Optimalität überprüfen. Ist $\mu_N \geq 0$, so ist x optimal. Ansonsten muss der Algorithmus bei Schritt 3 fortgesetzt werden.

(3) Bestimme einen Kandidaten $j_{m+p} \in N$, sodass $\mu_{j_{m+p}} < 0$.

(4) Berechne die Änderung der Basisvariablen $w = -\eta_B^p = A_B^{-1} A_{\bullet j_{m+p}}$. Ist $w \leq 0$, so ist das LP unbeschränkt und besitzt also keine Lösung. Der Algorithmus kann in diesem Fall hier abgebrochen werden.

(5) Bestimme den Basisindex j_q, der die Basis verlassen soll, durch $x(\theta) = x - \theta w \overset{!}{\geq} 0$. Hierfür definiert man

$$\hat{\theta} := \min_{1 \leq s \leq m} \left\{ \begin{array}{ll} \frac{x_{js}}{w_s} & , \text{ falls } w_s > 0 \\ \infty & , \text{ falls } w_s \leq 0 \end{array} \right\} = \frac{x_{j_q}}{w_q} \ .$$

(6) Nun kann man die Werte für den nächsten Iterationsschritt bestimmen:

$$x_{j_{m+p}}^{neu} = \hat{\theta} \ , \quad x_B^{neu} := x_B - \hat{\theta} w \quad (\text{speziell } x_{j_q} = 0) \ ,$$

$$B^{neu} = (B \backslash \{j_q\}) \cup \{j_{m+p}\} \ , \quad N^{neu} = (N \backslash \{j_{m+p}\}) \cup \{j_q\} \ , \quad A_B^{neu} = A_B + (A_{\bullet j_{m+p}} - A_{\bullet j_q}) e_q^T \ .$$

Nun kann man mit dem nächsten Iterationsschritt beginnen. Zu beachten ist hierbei nur, dass x_B^{neu} nicht mit dem x_B für den folgenden Iterationsschritt übereinstimmt. Dieses muss erst aus der neuen Basis B zusammengestellt werden.

Beispiele zum revidierten Simplex-Algorithmus

Beispiel 1 Sei das LP

$$\begin{array}{c} \min x_1 - x_2 \\ x_1 - x_2 \leq 2 \\ x_1 + x_2 \leq 6 \\ x \geq 0 \end{array} \quad \Leftrightarrow \quad \begin{array}{c} \min x_1 - x_2 \\ x_1 - x_2 + x_3 = 2 \\ x_1 + x_2 + x_4 = 6 \\ x \geq 0 \end{array}$$

gegeben. Um den ersten Iterationsschritt zu beginnen setzt man nun $B = \{3, 4\}$ und $N = \{1, 2\}$. Somit ist $A_B = I^{2 \times 2}$, $c_B^T = (0, 0),\ c_N^T = (1, -1),\ x_B^T = (2, 6)$ und $x_N^T = (0, 0)$.

IS 1) $\mu^T = c^T - c_B^T A_B^{-1} A = (1, -1, 0, 0)$. Also $\mu_2 < 0$. Wähle $p = 2$, also $j_{m+p} = 2$. Somit ist $w = A_B^{-1} A_{\bullet 2} = \binom{-1}{1}$. Da $w_1 < 0$ ergibt sich also die Schrittweite $\hat{\theta} = \frac{x_{j_2}}{w_2} = 6$. Somit ist also

$$x_B^{neu} = x_B - \theta w = \begin{pmatrix} 8 \\ 0 \end{pmatrix}, \quad x_N^{neu} = \begin{pmatrix} 0 \\ 6 \end{pmatrix}, \quad B^{neu} = \{3, 2\}, \quad N^{neu} = \{1, 4\}.$$

IS 2) Nach dem ersten Iterationsschritt ist man in der Ecke $x^T = (0, 6, 8, 0)$. Mit $B = \{3, 2\}$ erhält man nun $A_B = \begin{pmatrix} 1 & -1 \\ 0 & 1 \end{pmatrix}$, $c_B^T = (0, -1)$, $c_N^T = (1, 0)$ und $x_B = \binom{x_3}{x_2} = \binom{8}{6}$. Nun ist

$$\pi^T = c_B^T A_B^{-1} = (0, -1) \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} = (0, -1), \quad \text{also } \mu^T = c^T - \pi^T A = (2, 0, 0, 1).$$

Also ist μ und auch insbesondere $\mu_N \geq 0$ und somit sagt die Theorie, dass man bei x das Optimum erreicht hat.

Beispiel 2

$$\begin{aligned} \min -10x_1 + 57x_2 + 9x_3 + 24x_4 \\ 0.5x_1 - 5.5x_2 - 2.5x_3 + 9x_4 + x_5 = 0 \\ 0.5x_1 - 1.5x_2 - 0.5x_3 + x_4 + x_6 = 0 \\ x_1 + x_7 = 1 \\ x \geq 0 \end{aligned}$$

Input: $c^T = (-10, 57, 9, 24, 0, 0, 0)$, $B = \{5, 6, 7\}$, $N = \{1, 2, 3, 4\}$, $A_B = I^{3\times 3}$, $x_B^T = (0, 0, 1)$, $x_N^T = (0, 0, 0, 0)$.

1. IS: $\mu_N^T = (\boxed{\text{-10}}, 57, 9, 24)$ $\Rightarrow$ wähle $p = 1$, also $j_4 = 1$ „entering variable“

$$\Rightarrow \quad w = \begin{pmatrix} 0.5 \\ 0,5 \\ 1 \end{pmatrix}, \quad \theta = \begin{pmatrix} \boxed{0} \\ 0 \\ 1 \end{pmatrix}$$

$\Rightarrow$ wähle $q = 1$, also $j_1 = 5$ „leaving variable“.

2. IS: $B = \{1, 6, 7\}$, $N = \{5, 2, 3, 4\}$, $x^T = (0, 0, 0, 0, 0, 0, 1)$, $\mu_N^T = (20, \boxed{\text{-53}}, -41, 204)$
$\Rightarrow$ wähle $p = 2$, also $j_5 = 2$ „entering variable“

$$\Rightarrow \quad w = \begin{pmatrix} -11 \\ 4 \\ 11 \end{pmatrix}, \quad \theta = \begin{pmatrix} \infty \\ \boxed{0} \\ 0,09 \end{pmatrix}$$

$\Rightarrow$ wähle $q = 2$, also $j_2 = 6$ „leaving variable“.

3. IS: $B = \{1, 2, 7\}$, $N = \{5, 6, 3, 4\}$, $x^T = (0, 0, 0, 0, 0, 0, 1)$, $\mu_N^T = (6.75, 13.25, \boxed{\text{-14.5}}, 98)$
$\Rightarrow$ wähle $p = 3$, also $j_6 = 3$ „entering variable“

$$\Rightarrow \quad w = \begin{pmatrix} 0.5 \\ 0.5 \\ -0.5 \end{pmatrix}, \quad \theta = \begin{pmatrix} \boxed{0} \\ 0 \\ \infty \end{pmatrix}$$

$\Rightarrow$ wähle $q = 1$, also $j_1 = 1$ „leaving variable“.

4. IS: $B = \{3, 2, 7\},\ N = \{5, 6, 1, 4\},\ x^T = (0, 0, 0, 0, 0, 0, 1),\ \mu_N^T = (-15, 93, 29, \boxed{\text{-18}})$
$\Rightarrow$ wähle $p = 4$, also $j_7 = 4$ „entering variable“

$$\Rightarrow \quad w = \begin{pmatrix} -8 \\ 2 \\ 0 \end{pmatrix}, \quad \theta = \begin{pmatrix} \infty \\ \boxed{0} \\ \infty \end{pmatrix}$$

$\Rightarrow$ wähle $q = 2$, also $j_2 = 2$ „leaving variable“.

5. IS: $B = \{3, 4, 7\},\ N = \{5, 6, 1, 2\},\ x^T = (0, 0, 0, 0, 0, 0, 1),\ \mu_N^T = (\boxed{\text{-10.5}}, 70.5, 20, 9)$
$\Rightarrow$ wähle $p = 1$, also $j_4 = 5$ „entering variable“

$$\Rightarrow \quad w = \begin{pmatrix} 0.5 \\ 0.25 \\ 0 \end{pmatrix}, \quad \theta = \begin{pmatrix} \boxed{0} \\ 0 \\ \infty \end{pmatrix}$$

$\Rightarrow$ wähle $q = 1$, also $j_1 = 3$ „leaving variable“.

6. IS: $B = \{5, 4, 7\},\ N = \{3, 6, 1, 2\},\ x^T = (0, 0, 0, 0, 0, 0, 1),\ \mu_N^T = (21, \boxed{\text{-24}}, -22, 93)$
$\Rightarrow$ wähle $p = 2$, also $j_5 = 6$ „entering variable“

$$\Rightarrow \quad w = \begin{pmatrix} -9 \\ 1 \\ 0 \end{pmatrix}, \quad \theta = \begin{pmatrix} \infty \\ \boxed{0} \\ \infty \end{pmatrix}$$

$\Rightarrow$ wähle $q = 2$, also $j_2 = 4$ „leaving variable“.

7. IS: $B = \{5, 6, 7\}$ Man ist also wieder bei der Ausgangsbasis vom ersten Iterationsschritt gelandet. Möglich wird dies durch das Durchwandern von degenerierte Ecken. Man wechselt zwar bei jedem Schritt die Basis, jedoch nicht die Ecke. In diesem Beispiel sieht man dies gut daran, dass man in jedem Iterationsschritt die Basislösung $x^T = (0, 0, 0, 0, 0, 0, 1)$ erhält. In einem solchen falls sagt man „der Algorithmus kreist“.

Satz 3.13 Sei $\{\min c^T x,\ Ax = b,\ x \geq 0\}$ ein lineares Programm mit einer zulässigen Menge $P \neq \varnothing$. Seien alle Basislösungen nicht degeneriert, so findet man nach endlich vielen Schritten das Minimum, oder man stellt fest, dass das Problem unbeschränkt ist.

Beweis Sind alle Ecken in P nicht degeneriert, so durchläuft der Simplex-Algorithmus keine Ecke mehrfach. Da P nach Korollar 2.14 jedoch nur endlich viele Ecken besitzt, folgt die Aussage. ■

3.5 Die Gauß-LU-Zerlegung

3.5.1 Die LU-Zerlegung ohne Pivotisierung

Um ein Gleichungssystem der Art $Ax = b$ mit $A \in GL(m, \mathbb{R})$ vereinfacht lösen zu können, bietet einem das folgende Verfahren die Möglichkeit die Matrix A in eine untere Dreiecksmatrix (lower matrix) und in eine obere Dreiecksmatrix (upper matrix) zu zerlegen. Diese sind dabei von der Gestalt

$$L = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ * & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ * & \cdots & * & 1 \end{pmatrix} \quad \text{und} \quad U = \begin{pmatrix} * & \cdots & \cdots & * \\ 0 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & * \end{pmatrix} .$$

Somit kann das LGS

$$Ax = b \quad \Leftrightarrow \quad L \underbrace{Ux}_{=:y} = b$$

vereinfacht gelöst werden. Man bestimmt zuerst y durch $Ly = b$ und danach x durch $Ux = y$. Man hat zwar nun zwei LGS zu lösen, der Rechenaufwand nimmt jedoch trotzdem ab, da die Werte für y und danach x fast direkt abgelesen werden können, ohne die zwei Matrizen L und U noch umformen zu müssen.

Algorithmus: Durch den Algorithmus der LU-Zerlegung wird die Matrix A Schritt für Schritt in die Matrix U überführt. Im ersten Schritt muss also das Ziel sein, in der ersten Spalte von $A =: A_{(1)}$ ab dem zweiten Element Nullen zu schaffen. Hierfür berechnet man zuerst die Quotienten

$$l_{i,1} := \frac{a_{i,1}}{a_{1,1}} .$$

Danach zieht man jeweils das $l_{i,1}$-te der ersten Zeile von der i-ten Zeile ab. Hat man dies für $i = 2, ..., m$ durchgeführt, besteht die erste Spalte von $A_{(1)}$ bis auf das oberste Element wie gewünscht nur noch aus Nullen. Genau diese Operationen lassen sich nun durch Multiplikation einer unteren Dreiecksmatrix der Form

$$I - \begin{pmatrix} 0 \\ l_{2,1} \\ \vdots \\ l_{m,1} \end{pmatrix} e_1^T = \begin{pmatrix} 1 & & & \\ -l_{2,1} & \ddots & & \\ \vdots & & \ddots & \\ -l_{m,1} & & & 1 \end{pmatrix} =: L_{(1)}^{-1}$$

ausdrücken. Man erhält

$$L_{(1)}^{-1} \cdot A_{(1)} = \left(\begin{array}{c|ccc} a_{11} & a_{12} & \cdots & a_{1n} \\ \hline 0 & & & \\ \vdots & & \widetilde{A_{(2)}} & \\ 0 & & & \end{array}\right) =: A_{(2)} .$$

Iterativ folgt nun für $A_{(2)}$ der selbe Schritt, d.h. man zieht ein Vielfaches der zweiten Zeile von allen

darunter liegenden Zeilen ab. Für den j-ten Schritt bestimmt man deshalb allgemein die Quotienen

$$l_{i,j} = \frac{A^{(j)}_{i,j}}{A^{(j)}_{j,j}} \quad \text{für } i = j+1, ..., m$$

und definiert sich $l^{(j)} \in \mathbb{R}^m$ mit

$$l^{(j)}_i := \begin{cases} 0 & \text{, für } i \leq j \\ l_{i,j} & \text{, für } i > j \end{cases} .$$

Somit ergibt sich nun mit $L^{-1}_{(j)} := I - l^{(j)} e_j^T$ der j-te Schritt:

$$L^{-1}_{(j)} \cdot A_{(j)} = L^{-1}_{(j)} \cdot \ldots \cdot L^{-1}_{(1)} \cdot A_{(1)} = A_{(j+1)} .$$

Nach genau $m-1$ Schritten hat nun A die Gestalt einer oberen Dreiecksmatrix und man definert $A_{(m)} =: U$. Die zugehörige untere Dreiecksmatrix L erhält man nun recht einfach, da

$$L := \left(L^{-1}_{(m-1)} \cdot \ldots \cdot L^{-1}_{(1)}\right)^{-1} = L_{(1)} \cdot \ldots \cdot L_{(m-1)} = \begin{pmatrix} 1 & & & & \\ l_{2,1} & 1 & & & \\ l_{3,1} & l_{3,2} & \ddots & & \\ \vdots & \vdots & \ddots & \ddots & \\ l_{m,1} & l_{m,2} & \cdots & l_{m,m-1} & 1 \end{pmatrix}$$

genau die gewünscht Form besitzt und nun gilt $A = L \cdot U$.

Beispiel Sei $A = \begin{pmatrix} 3 & 1 & 6 \\ 2 & 1 & 3 \\ 1 & 1 & 1 \end{pmatrix} =: A_{(1)}$ gegeben.

$$\text{setze} \quad L^{-1}_{(1)} := \begin{pmatrix} 1 & 0 & 0 \\ -\frac{2}{3} & 1 & 0 \\ -\frac{1}{3} & 0 & 1 \end{pmatrix} \quad \Rightarrow \quad L^{-1}_{(1)} A_{(1)} = \begin{pmatrix} 3 & 1 & 6 \\ 0 & \frac{1}{3} & -1 \\ 0 & \frac{2}{3} & -1 \end{pmatrix} =: A_{(2)}$$

$$\text{setze} \quad L^{-1}_{(2)} := \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -2 & 1 \end{pmatrix} \quad \Rightarrow \quad L^{-1}_{(2)} A_{(2)} = \begin{pmatrix} 3 & 1 & 6 \\ 0 & \frac{1}{3} & -1 \\ 0 & 0 & 1 \end{pmatrix} =: A_{(3)} =: U$$

Schließlich setzt man $L := L_{(1)} \cdot L_{(2)}$. Somit gilt nun

$$LU = \begin{pmatrix} 1 & 0 & 0 \\ \frac{2}{3} & 1 & 0 \\ \frac{1}{3} & 2 & 1 \end{pmatrix} \begin{pmatrix} 3 & 1 & 6 \\ 0 & \frac{1}{3} & -1 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 3 & 1 & 6 \\ 2 & 1 & 3 \\ 1 & 1 & 1 \end{pmatrix} = A .$$

3.5.2 Die LU-Zerlegung mit Pivotisierung

Bei der Berechnung der LU-Zerlegung einer Matrix muss man um die Vektoren $l^{(j)}$ für $1 \leq j < m$ zu bestimmen in jedem Schritt eine Division durch das Pivotelement $A^{(j)}_{j,j}$ durchführen. Bisher wurde ohne Einschränkung angenommen, dass alle Pivotelemente von 0 verschieden waren. Dies gilt jedoch nicht allgemein.
In einem solchen Fall verschafft man sich Abhilfe durch das Vertauschen der Zeilen von A, einer sog. *Spaltenpivotisierung*. Im j-ten Schritt der Zerlegung bestimmt man hierfür

$$\max_{k=j}^{m} |A^{(j)}_{k,j}| =: |A^{(j)}_{k',j}| \ .$$

Ist $|A^{(j)}_{k',j}| = 0$, so ist die Matrix A singulär und die LU-Zerlegung kann nicht durchgeführt werden. Ansonsten tauscht man die Zeilen k' und j miteinander. Dies geschieht durch die Multiplikation von rechts mit einer sog. Permutationsmatrix P der Form

$$P = (e_1, \ldots, e_{j-1}, e_{k'}, e_{j+1}, \ldots, e_{k'-1}, e_j, e_{k'+1}, \ldots, e_m) \ ,$$

also einer Einheitsmatrix, in der genau die Spalten k' und j vertauscht sind. Wie auch die Einheitsmatrix sind diese Permutationsmatrizen selbstinvers. Nach dem Vertauschen der Spalten kann mit der LU-Zerlegung fortgefahren werden und wie bei der LU-Zerlegung ohne Pivotisierung ergibt sich nach dem $m-1$-ten Schritt

$$A = \underbrace{P_{(1)}L_{(1)} \cdot \ldots \cdot P_{(m-1)}L_{(m-1)}}_{=:L} \cdot R \ .$$

Zu beachten ist nur, dass die Matrix L nun keine Dreicksform, jedoch trotzdem eine für das Lösen eines linearen Gleichungssystems einfache Form besitzt.

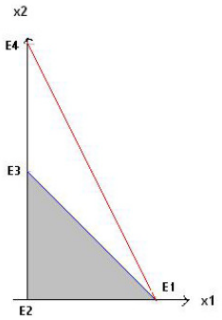
3.6 Degeneriertheit und Schleifen

Durchläuft der Simplex-Algorithmus eine degenerierte Ecke, so könnte das Problem auftreten, dass durch einen Basiswechsel zwar zu einer neuen Basis, jedoch keiner neuen Ecke gelangt. Dies ist genau dann der Fall, wenn die Schrittweite $\hat{\theta} = 0$ gewählt wurde. Dieses Problem hätte zur Folge, dass der Simplexalgorithmus kreist und sich somit in einer Endlosschleife befindet.

Beispiele

a) Im $\mathbb{R}^2$ gibt es keine „echte“ Degeneriertheit, sondern nur wegen den redundanten Ungleichungen.

$$\begin{array}{rrcl} & x_1 + x_2 & \leq & 1 \\ \text{redundant} \rightarrow & 2x_1 + x_2 & \leq & 2 \\ & x_1, x_2 & \geq & 0 \end{array} \quad \Leftrightarrow \quad \begin{array}{rcl} x_1 + x_2 + x_3 & = & 1 \\ 2x_1 + x_2 + x_4 & = & 2 \\ x & \geq & 0 \end{array}$$



Für die Ecken gibt es dann folgende Basislösungen:

$$\begin{array}{lllll} E_1 & x=(1,0,0,0)^T & B=\{1,2\} & N=\{3,4\} & \text{degeneriert} \\ & & B=\{1,3\} & N=\{2,4\} & \text{degeneriert} \\ & & B=\{1,4\} & N=\{2,3\} & \text{degeneriert} \\ E_2 & x=(0,0,1,2)^T & B=\{3,4\} & N=\{1,2\} & \\ E_3 & x=(0,1,0,1)^T & B=\{2,4\} & N=\{1,3\} & \\ E_4 & x=(0,2,-1,0)^T & B=\{2,3\} & N=\{1,4\} & \text{unzulässig} \end{array}$$

b) Im $\mathbb{R}^3$ hingegen gibt es echte Degeneriertheit. Dies ist bei Ecken der Fall, die von mehr als drei Ungleichungen beschrieben werden. Zum Beispiel der Gipfelpunkt einer vierseitigen Pyramide.

c) *capacitated transport problem*

$$P_K = \left\{ x_{ij} \;\middle|\; 1 \le i,j \le k,\ \sum_{j=1}^{k} x_{ij} = 1\ \forall i,\ \sum_{j=1}^{k} x_{ij} = 1\ \forall j,\ x_{ij} \ge 0 \right\}$$

Dieser Polyeder hat $k!$ Ecken und $2^{k-1}k^{k-2}$ Basen für die Ecken. Im Fall $k = 8$ bedeutet dies 40320 Ecken und 33554432 Basen.

3.6.1 Anti-Schleifen-Methoden

Um das oben beschriebene Problem zu beheben, gibt es mehrere Methoden.

1. **Bland-Methode** von 1977. Diese Methode basiert auf einer guten theoretischen Grundlage, lässt sich in der Praxis aber auf Grund des hohen Rechenaufwandes schlecht anwenden. Die Methode funktioniert folgendermaßen: Wähle den kleinsten Index $j \in N$ mit $\mu_j < 0$ als „entering variable“ und analog den kleinsten Index $B_i \in B$ mit $\frac{x_{B_i}}{w_i} =: \hat{\theta}$ als „leaving variable“.

2. **Perturbations- (Störungs-) Ansatz** Sei $\tilde{\varepsilon}^T := (\varepsilon_1, \ldots, \varepsilon_n),\ \tilde{\varepsilon} > 0$ der Störungsvektor mit $\tilde{\varepsilon} \in \mathbb{R}^m$, so wird das LP durch LP_ε mit $\{\min c^T x,\ Ax = b + \tilde{\varepsilon},\ x \ge 0\ \}$ ersetzt.

   Wie funktioniert nun der Perturbationsansatz?

   Sei $b(\varepsilon) = b + E\eta$, mit $\eta = (\varepsilon, \varepsilon^2, \ldots, \varepsilon^m)^T$, $0 < \varepsilon \in \mathbb{R}$ und E regulär. Sei weiter $A_B x_B(\varepsilon) = b(\varepsilon)$, so ergibt sich

   $$(*) \quad x_B(\varepsilon) = x_B + A_B^{-1}E\eta = x_B + \sum_{k=1}^{m} (A_B^{-1}E)_{\bullet k}\ \varepsilon^k \ .$$

   Wähle nun die Matrix E so, dass $\sum_{k=1}^{m} (A_B^{-1}E)_{\bullet k}\varepsilon^k > 0\ \forall \varepsilon$. Wann könnte es nun passieren, dass $\forall \varepsilon$ gilt

   $$\frac{(x_B(\varepsilon))_i}{y_i} = \frac{(x_B(\varepsilon))_l}{y_l} \ ,$$

   also $\theta_i = \theta_l$? Dies ist nur dann der Fall, wenn nach (*)

   $$\frac{(A_B^{-1}E)_{ik}}{y_i} \ = \ \frac{(A_B^{-1}E)_{lk}}{y_l} \quad \forall\ k = 1, \ldots, m \ .$$

Dies würde aber wiederum bedeuten, das die Zeilen i und l in A_B^{-1} linear abhängig wären! Das ist aber nach Voraussetzung nicht möglich, da zu jeder Basislösung die Matrix A_B^{-1} regulär ist. Somit folgt also, dass die Schrittweiten nicht identisch sind. Daher ist also der „leaving index“ eindeutig.

3. **Anti-Cycling-Strategy** Diese Strategie beruht auf dem Störungsansatz, verwendet allerdings eine eindeutige Matrix. Hier wird nämlich die Matrix $E = A_B$ gesetzt. Dann ist $x_B(\varepsilon) = x_B + \eta > x_B$, da $\varepsilon > 0$. Daraus folgt aber, dass auch immer $\hat{\theta} > 0$ ist.

Satz 3.14 (Perturbationsmethode)

(a) Das Lineare Problem ist genau dann lösbar, wenn das $LP(\varepsilon)$ für jedes $\varepsilon > 0$ lösbar ist.

(b) Es gibt $\varepsilon_0 > 0$ derart, sodass $\forall 0 < \varepsilon < \varepsilon_0$ jede zulässige Basis von LP(ε) eine nicht degenerierte Basislösung von LP(ε) bestimmt und gleichzeitig eine zulässige Basislösung des LPs ist. Des Weiteren bestimmt die optimale Basis von LP(ε) auch die optimale Basis von LP.

3.7 Die Phase I

Bisher war es durch das Einfügen von Schlupfvariablen recht einfach, eine zulässige Basislösung für den Start des Simplex-Algorithmus zu finden. Im Folgenden ist es nun das Ziel auch für ein Lineares Programm der Form $\{\min c^T x,\ Ax = b,\ x \geq 0\}$ eine zulässige Basislösung zu finden.
Dies führt auf das **Phase-I-Problem**:

$$\min \sum_{i=1}^{m} y_i, \quad Ax + y = b,\ x, y \geq 0$$

1. $A' = (A, I)$ hat vollen Zeilenrang m
2. Es sind mehr Spalten als Zeilen vorhanden
3. Setzt man $x = 0$ und $y = b \geq 0$, so ist $B = \{n+1, n+2, \ldots, n+m\}$ eine Basis und $\binom{x}{y}$ eine zulässige Basislösung. Die y_i bezeichnet man hierbei als künstliche Variablen und die x_i als Strukturvariablen.
4. $A'_B = A'^{-1}_B = I$

Anwendung des Simplexverfahrens auf das Phase-I-Problem

Sei $z^* = \binom{x^*}{y^*}$ eine Lösung.

(1) Falls $\sum_{i=1}^{m} y_i^* > 0 \quad \Rightarrow \quad P = \{x \mid Ax = b,\ x \geq 0\} = \varnothing$

(2) Falls $\sum_{i=1}^{m} y_i^* = 0 \quad \Leftrightarrow \quad y_i^* = 0, \quad i = 1, \ldots, m$

(2a) Ist $B^* \cap \{n+1, \ldots, n+m\} = \varnothing$, so startet man die Phase II mit $B = B^*$ und $x = x^*$.

(2b) Ist $B^* \cap \{n+1, \ldots, n+m\} = \{i_1, \ldots, i_t\},\ t \geq 1$, so entfernt man nach und nach alle künstlichen Variablen aus der Basis. Dies ist möglich, da wegen $y^* = 0$ die Basis degeneriert ist.

Betrachte $A_{\bullet j}$, wobei $j \in \{1, \ldots, n\} \cap N$, und $w = (A'_B)^{-1} A_{\bullet j}$ mit $w_i \neq 0$, für $\{i_1, \ldots, i_t\}$, so ist $\hat{\theta} = 0$. Nun führt man eine Simplexiteration mit x_j als „entering-variable“und y_{B_i} als „leaving-variable“ durch. Dies wiederholt man so lange, bis alle künstlichen Variablen die Basis verlassen haben, d.h. man bei Fall (2a) landet oder bis $((A'_B)^{-1} A_j)_i = 0\ \forall\ i \in B^* \cap \{n+1, \ldots, n+m\},\ j \in N \cap \{1, \ldots, n\}$.

Notation $B_x = B \cap \{1, \ldots, n\}\ , \quad B_y = B \cap \{n+1, \ldots, n+m\}$

$$z_B^* = (A'_B)^{-1} b - (A'_B) A'_N z_N^* = \begin{pmatrix} x^*_{B_x} \\ y^*_{B_y} \end{pmatrix} = \begin{pmatrix} * \\ 0 \end{pmatrix} - \left(\begin{array}{c|c} * & * \\ \hline 0 & * \end{array}\right) \begin{pmatrix} x_N^* \\ y_N^* = 0 \end{pmatrix} = \begin{pmatrix} x^*_{B_x} \\ 0 \end{pmatrix}$$

Für beliebige x_N^* ist der untere Teil des Gleichungssystems immer erüllt. Daraus folgt, dass zu B_y gehörende Zahlen redundant sind. Diese Zeilen müssen somit eliminiert werden. Basis für die Phase II ist $B_x,\ \tilde{A} = A,\ x^*_{B_x} \geq 0$ ist eine zulässige Basislösung.

Bemerkung

1. Die Simplex-Methode ist daher Phase I + Phase II, mit der Phase II $\{\min c^T x,\ Ax = b,\ x \geq 0\}$ und Œ $b \geq 0$.

2. In der Praxis werden die redundanten Zeilen (in der Regel) nicht entfernt.

Beispiel

$$\begin{aligned} x_1 + 2x_2 &= 2 \\ x_1 - x_2 &= 1 \quad , \ x \geq 0 \end{aligned}$$

Phase I:

$$\begin{aligned} \min x_3 + x_4 \qquad & x \geq 0 \\ x_1 + 2x_2 + x_3 &= 2 \\ x_1 - x_2 + x_4 &= 1 \qquad A' = \begin{pmatrix} 1 & 2 & 1 & 0 \\ 1 & -1 & 0 & 1 \end{pmatrix} \end{aligned}$$

1. $c^T = (0,0,1,1),\ B = \{3,4\},\ N = \{1,2\},\ A_B = I^{2\times 2},\ x_B^T = (x_3, x_4) = (2,1),\ x_N^T = (0,0).$

$$\mu_N^T = c_N^T - c_B^T A_B^{-1} A_N = (0,0) - (1,1)\begin{pmatrix} 1 & 2 \\ 1 & -1 \end{pmatrix} = (\boxed{-2}, -1)$$

$\Rightarrow$ wähle x_1 als „entering variable“

$$w = A_B^{-1} A_{\bullet 1} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \quad x_B^{neu} = x_B - \theta w = \begin{pmatrix} 2 \\ 1 \end{pmatrix} - \theta \begin{pmatrix} 1 \\ 1 \end{pmatrix} \overset{!}{\geq} 0 \Rightarrow\ \hat{\theta} = \theta_2 = 1$$

$\Rightarrow$ wähle x_4 als „leaving variable“

2. $B = \{3,1\},\ N = \{4,2\},\ x_B^{Neu} = \binom{2}{1} - 1\binom{1}{1} = \binom{1}{0},\ x_N^{Neu} = \binom{0}{0} + 1\binom{1}{0} = \binom{1}{0},\ x^T = (1,0,1,0),$ $A_B = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$

$$\pi^T = c_B^T A_B^{-1} \Leftrightarrow \pi^T A_B = c_B^T \Rightarrow A_B^T \pi = c_B \rightarrow \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}\begin{pmatrix} \pi_1 \\ \pi_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \Rightarrow \pi^T = (1,-1)$$

$$\mu_N^T = (1,0) - (1,-1)\begin{pmatrix} 0 & 2 \\ 1 & -1 \end{pmatrix} = (2, \boxed{-3})$$

$\Rightarrow$ wähle x_2 als „entering variable“

$$A_B w = A_{\bullet 2} \Rightarrow\ \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} w = \begin{pmatrix} 2 \\ -1 \end{pmatrix} \Rightarrow\ w_1 = 3,\ w_2 = -1$$

$$x_B^{Neu} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} - \theta \begin{pmatrix} 3 \\ -1 \end{pmatrix} \overset{!}{\geq} 0\ \Rightarrow\ \hat{\theta} = 1/3$$

$\Rightarrow$ wähle x_3 als „leaving variable“

$$x_B^{neu} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} - \frac{1}{3}\begin{pmatrix} 3 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1/3 \end{pmatrix}, \quad x_N^{Neu} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} + \frac{1}{3}\begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

3. $B = \{2,1\},\ N = \{4,3\},\ x^T = (\frac{4}{3}, \frac{1}{3}, 0, 0)\ A_B = \begin{pmatrix} 2 & 1 \\ -1 & 1 \end{pmatrix}$

$$\Rightarrow\ \mu_n^T = c_N^T - \pi^T A_N = c_N^T = (1,1)$$

Stop! Optimum der Phase I erreicht und es gilt $B \cap \{3,4\} = \varnothing$. Deshalb kann man nun die Phase II mit $B = \{2,1\},\ x_B = (\frac{4}{3}, \frac{1}{3}) \geq 0$ beginnen.

3.7.1 Die M-Methode

Die M-Methode (sprich: Groß M Methode) wird in der Literatur als „Big M Approach“ bezeichnet. Sie kombiniert die Phase I und die Phase II mit einem genügend großen $M \in \mathbb{R}$ zu folgendem Ansatz:

$$\min \sum_{j=1}^{n} c_j x_j + M \sum_{i=1}^{m} y_i \quad \text{unter} \quad Ax + y = b,\ x, y \geq 0$$

Sollte als Startwert beispielsweise $M = 10^2$ nicht genügen, vergrößert man diesen Wert zu $M' = 10^2 M$ und dann immer so weiter.

3.8 Pricing-Strategien

Um die Leistung des Simplex-Algorithmus zu erhöhen, ist man danach bestrebt ein lineares Programm mit möglichst wenigen Iterationsschritten zu lösen. Die Anzahl der benötigten Iterationsschritte lässt sich in jedem Schritt durch eine bestimmte Wahl der „entering variable“ beeinflussen. Dies ist natürlich nur möglich, wenn man während eines Iterationsschrittes mehrere Möglichkeiten hierfür hat, d.h. der Vektor der reduzierten Kosten mehrere negative Einträge besitzt.

Notation Sei $B = \{j_1, \ldots, j_m\}$ und $N = \{j_{m+1}, \ldots, j_{m+n}\}$.

Die bekanntesten Pricing-Strategien sind:

(i) **kleinster Index:** Man wählt $j_p \in N$ mit $p = \min\{k \mid \mu_{j_k} < 0\}$ als den in die Basis wandernden Index.

(ii) **kleinster Variablen Index:** Man wählt $j_p \in N$ mit $j_p = \min\{j_k \mid \mu_{j_k} < 0\}$ als den in die Basis wandernden Index.

(iii) **Steilster Abstieg (Dantzig-Regel):** In die Basis wandert der Index j_p mit

$$\mu_{j_p} = \min\{\mu_{j_k} \mid j_k \in N,\ \mu_{j_k} < 0\}\ .$$

Diese Regel wird in der Realität häufig angewandt, kann jedoch bei großen n sehr rechenaufwendig werden. Um dem Abhilfe zu verschaffen verwendet man bei großen n oft das sog. „partial pricing“. Hierbei wird jeweils nur eine Teilmenge der Nichtbasisvariablen betrachtet und auf diese die Danzig-Regel angewandt.

(iv) **Steepest-Edge-Pricing:** Als „entering variable“ wählt man die Variable (geom. Kante), die am steilsten bzgl. der Zielfunktion fällt, d.h. die pro Einheit den größten Fortschritt in der Zielfunktion bewirkt. mathematisch:

$$\begin{pmatrix} x_B \\ x_N \end{pmatrix}^{neu} = \begin{pmatrix} x_B \\ x_N \end{pmatrix} + \hat{\theta}\eta^p \quad \text{mit } \eta^p = \begin{pmatrix} -A_B^{-1}A_{\bullet j_p} \\ e_p \end{pmatrix} \text{ für } j_p \in N$$

reduzierte Kosten:

$$\mu_{j_p} = c_{j_p} - c_B^T A_B^{-1} A_{\bullet j_p} = c^T \eta^p$$

Nun wählt man $j_p \in N$ mit

$$\frac{c^T\eta^p}{\|\eta^p\|_2} = \min_{j_k \in N} \frac{c^T\eta^k}{\|\eta^k\|_2} \ .$$

Geometrisch interpretiert wählt man diejenige Kante, mit der der Winkel zwischen Kante und Antigradient $(-c)$ möglichst klein ist.

Der Vorteil dieser Strategie liegt darin, dass man deutlich weniger Iterationsschritte für das Lösen eines linearen Programms benötigt. Trotzdem ist sie nicht immer die Beste, da man in jedem Schritt wegen des Lösens mehrerer Gleichungssyteme einen sehr hohen Rechenaufwand benötigt. Abhilfe hierfür schaffen sogenannte Update Formeln für $\|\eta^p\|_2^2$ von Goldfarb & Reid. Eine weitere Alternative ist das sog. *DEVEX-Pricing*, bei dem die Normen nur approximiert werden.

(v) **Best Progress:** Man berechnet für alle $\mu_{j_k} < 0$ auch die Schrittweiten $\hat{\theta}_{j_k}$ und wählt dann $j_p \in N$ mit

$$\underbrace{\hat{\theta}_{j_p}\mu_{j_p}}_{\Delta(c^Tx)} = \min\{\hat{\theta}_{j_k}\mu_{j_k} \mid j_k \in N,\ \mu_{j_k} < 0\}$$

als den in die Basis wandernden Index.

3.9 Laufzeiten des Simplex-Verfahrens

Bekanntlicher Weise ist die Laufzeit des Simplex-Algorithmus immer endlich. Auch bei Degeneriertheit ist dies durch bekannte Zusatzregeln erreichbar. Nun stellt sich die Frage, ob eine optimale *Pricing-Strategie* existiert, mit der der Simplex-Algorithmus für jedes lineare Problem polynomiale Laufzeiten in n und m besitzt.

Die Antwort hierauf ist leider NEIN. Wie sich in vielen Versuchen gezeigt hat, existieren für jede bekannte Pricing-Strategie Beispiele, für die der Simplex-Algorithmus mit der ausgewählten Strategie eine eher exponentielle Laufzeit besitzt. Die Optimalität einer Pricing-Strategie ist also von der Wahl des linearen Programms abhängig.

Um die Optimalität der Strategie des steilsten Abstiegs zu widerlegen, gibt es z.B. das lineare Programm

$$\max \sum_{j=1}^{n} 10^{n-j} x_j \qquad \text{unter} \quad 2\sum_{j=1}^{i-1} 10^{i-j} x_j + x_i \leq 100^{i-1} \ , \ i = 1, ..., n \ .$$

3.10 Spezielle Strukturen, obere und untere Schranken

Gegeben sei das Lineare Problem $\min c^T x$ unter $Ax = b$, wobei $l \leq x \leq u$ und $l < u$. Somit ist l die untere Schranke und u die obere Schranke für den Bereich der Basislösungen. Zur Anwendung gibt es zwei verschiedene Varianten.

1.) In der ersten Variante werden die Beschränkungen für die Strukturvariablen durch das Einfügen zusätzlicher Restriktionen bewirkt. Dies ist aber auch gleich der Grund, warum dieses Verfahren in der Realität nicht angewandt wird. Der Rechenaufwand wird einfach zu groß. Außerdem ist sie numerisch nicht sinnvoll, da die Rechenfehleranfälligkeit zu hoch wird. Das oben gegebene Problem ist äquivalent zu den Restriktionen $\tilde{x} := x - l \geq 0, \ x \leq u$. Somit gilt für $x = \tilde{x} + l$

$$\begin{array}{ccc} Ax = A\tilde{x} + Al = b & \Leftrightarrow & A\tilde{x} = b - Al =: \tilde{b} \\ x \leq u & \Leftrightarrow & \tilde{x} \leq u - l \end{array}$$

Mit der Schlupfvariable $z \in \mathbb{R}^n$ folgt nun für $z \geq 0$: $\quad \tilde{x} + z = u - l$

$$\text{Also insgesamt :} \qquad c \leftarrow \begin{pmatrix} c \\ 0 \end{pmatrix}, \quad x \leftarrow \begin{pmatrix} \tilde{x} \\ z \end{pmatrix}, \quad A \leftarrow \begin{pmatrix} A & 0 \\ I & I \end{pmatrix}, \quad b \leftarrow \begin{pmatrix} b - Al \\ u - l \end{pmatrix}$$

2.) In der 2. Variante wird der Simplex-Algorithmus an das LP $\{\min c^T x, \ Ax = b, \ l \leq x \leq u\}$ angepasst. Sei $B := \{j_1, \ldots, j_m\} \subset \{1, \ldots, n\}$ eine Basis, also $A_B = (A_{\bullet j_1}, \ldots, A_{\bullet j_m})$ invertierbar und $N := \{1, \ldots, n\} \backslash B = \{N_1, \ldots, N_{n-m}\}$. Die Nichtbasisindizes trennt man nun noch weiter in

$$N = N^+ \cup N^- \ , \quad N^+ = \{N_i \mid x_{N_i} = l_{N_i}\} \ , \quad N^- = \{N_i \mid x_{N_i} = u_{N_i}\} \ ,$$

da man einen Teil der Nichtbasisvariabeln auf die obere Grenze und einen Teil auf die untere Grenze setzt. Die Basislösung $x = \binom{x_B}{x_N}$ besimmt man somit nun durch

$$x_{N_i} := \begin{cases} l_{N_i} & , \ N_i \in N^+ \\ u_{N_i} & , \ N_i \in N^- \end{cases}$$

und schließlich x_B aus

$$A_B x_B + A_N x_N = b \quad \Rightarrow \quad x_B = A_B^{-1}(b - A_N x_N) \ .$$

Die Basislösung x ist nun *zulässig*, falls $l_B \leq x_B \leq u_B$ und *nicht degeneriert*, falls $l_B < x_B < u_B$.

Update des Simplex-Algorithmus:

Input: Sei x eine zulässige Basislösung und $N = N^+ \cup N^-$.

Schritt 1: Berechne die Multiplikatoren π aus $A_B^T\pi = c_B$ und die reduzierten Kosten: $\mu_N^T = c_N^T - \pi^T A_N$.

Schritt 2: Prüfe Optimalität: Hierfür betrachtet man zunächst $x_B = A_B^{-1}(b - A_N x_N)$ und

$$c^T x = c_B^T x_B + c_N^T x_N = c_B^T A_B^{-1} b + c_N^T x_N - c_B^T A_B^{-1} A_N x_N = \pi^T b + \mu_N^T x_N \ .$$

Sei nun y zulässig, d.h. $Ay = b,\ l \le y \le u$, so ist $c^T y = \pi^T b + \mu_N^T y_N$ und somit

$$c^T y - c^T x = \mu_N^T (y_N - x_N) = \mu_{N^+}^T (y_{N^+} - l_{N^+}) + \mu_{N^-}^T (y_{N^-} - u_{N^-}) \ .$$

y ist genau dann optimal, falls nun $\forall\ x$ gilt $c^T y \ge c^T x$, also $c^T y - c^T x \ge 0$. Dies ist genau dann der Fall, wenn

$$\mu_{N^+} \ge 0 \text{ (da } y_{N^+} \ge l_{N^+}) \text{ und } \mu_{N^-} \le 0 \text{ (da } y_{N^-} \le u_{N^+}) \ .$$

In diesem Fall: **STOP!** x ist Optimallösung! Ansonsten bestimmt man N_P aus $\{N_z \mid \mu_{N_z} < 0,\ N_z \in N^+,\ \mu_{N_z} > 0,\ N_z \in N^-\}$.

Schritt 3: Bestimmen der „entlangzuwandernden“ Kante: Berechne $\bar{x} = x + \theta\Delta x$. Da natürlich auch $\bar{x}$ wieder zulässig sein muss, also $A\bar{x} = b$, muss folgen

$$\theta A\Delta x = 0 \quad \Rightarrow \quad A_B\Delta x_B + A_N\Delta x_N = 0 \quad \Rightarrow \quad \Delta x_B = -A_B^{-1}A_N\Delta x_N \ .$$

Die Frage ist nun also nur noch, wie man das Δx_N am Besten wählt. $\forall\ N_i \in N$ mit $N_i \ne N_p$ setzt man $\Delta x_{N_i} = 0$.

1. Ist $N_p \in N^+$, also $x_{N_p} = l_{N_p}$, so wählt man $\Delta x_{N_p} = 1$
2. Ist $N_p \in N^-$, also $x_{N_p} = u_{N_p}$, so wählt man $\Delta x_{N_p} = -1$

Demnach wählt man also $\Delta x_{N_p} = -\text{sign}\mu_{N_p}$. Für Δx_B ergibt sich somit

$$\Delta x_B = -A_B^{-1}A_N e_p(-\text{sign}\mu_{N_p}) = \text{sign}\mu_{N_p}A_B^{-1}A_{\bullet N_p}$$

Nun berechne w aus $A_B w = A_{\bullet N_p}$. Es folgt:

$$\Delta x_B = \text{sign}\mu_{N_p} w,\ \Delta x_N = -\text{sign}\mu_{N_p} e_p$$

Schritt 4: Berechne die Schrittweite $\hat{\theta}$, sodass $l \le \bar{x} = x + \theta\Delta x \le u$

1. $N_P \in N^+, \quad \bar{x}_{N_p} = l_{N_p} + \theta \cdot 1 \le u_{N_p} \quad \Rightarrow \theta_{N_p} = \max,\ \text{sodass } \bar{x}_{N_p} \le u_{N_p} \Rightarrow\ \theta_{N_p} = u_{N_p} - l_{N_p}$
2. $N_P \in N^-, \quad \bar{x}_{N_p} = u_{N_p} + \theta \cdot (-1) \ge l_{N_p} \quad \Rightarrow \theta_{N_p} = u_{N_p} - l_{N_p}$
3. θ sodass $l_B \le x_B + \theta\Delta x_B \le u_B$

a) Falls $\Delta x_{B_i} < 0 \quad \Rightarrow l_B = x_{B_i} - \theta_i \Delta x_{B_i} \quad \Rightarrow \theta_i = \frac{l_{B_i} - x_{B_i}}{\Delta x_{B_i}}$

b) Falls $\Delta x_{B_i} > 0 \quad \Rightarrow u_B = x_{B_i} + \theta_i \Delta x_{B_i} \quad \Rightarrow \theta_i = \frac{u_{B_i} - x_{B_i}}{\Delta x_{B_i}}$

c) Falls $\Delta x_{B_i} = 0 \quad \Rightarrow \theta_i = \infty$

$$\Rightarrow \quad \hat{\theta} = \min\{u_{N_p} - l_{N_p}; \theta_i,\ B_i \in B\}$$

Schritt 5: Basis-Update

1. Fall $\hat{\theta} = u_{N_p} - l_{N_p}$:
   $N^+ \leftarrow N^+ \cup N_p, \quad N^- \leftarrow N^- \backslash N_p \quad$ falls $N_p \in N^-$.
   Dies würde bedeuten, dass das x_{N_p} von der oberen auf die untere Grenze springt.
   $N^+ \leftarrow N^+ \backslash N_p, \quad N^- \leftarrow N^- \cup N_p$ falls $N_p \in N^+$
   $B \leftarrow B\ , \quad x \leftarrow x + \hat{\theta}\Delta x$

2. Fall $\hat{\theta} = \theta_q$:
   $B \leftarrow (B \backslash B_q) \cup N_p, \quad N^+ \leftarrow N^+ \cup B_q$, falls $\Delta x_{B_q} < 0\ , \quad N^- \leftarrow N^- \cup B_q$, falls $\Delta x_{B_q} > 0$
   $x \leftarrow x + \hat{\theta}\Delta x$

3. $\hat{\theta} = \infty \quad$ Stopp, das LP ist unbeschränkt. Sonst gehe wieder zu Schritt 1.

Beispiel

$$\begin{aligned} &\min x_1 - 2x_2 - x_3 - x_4 + 4x_5 \qquad x^T = (x_1, x_2, x_3, x_4, x_5) \\ &-5x_2 - 2x_3 + 4x_4 - 6x_5 = -8 \\ &x_1 + 4x_2 + x_3 - 3x_4 + 5x_5 = 12 \qquad A = \begin{pmatrix} 0 & -5 & -2 & 4 & -6 \\ 1 & 4 & 1 & -3 & 5 \end{pmatrix} \end{aligned}$$

$$\begin{aligned} 0 \le x_1 \le 6 \\ 0 \le x_2 \le 2 \\ 0 \le x_3 \le 10 \\ 0 \le x_4 \le 10 \\ 0 \le x_5 \le 3 \end{aligned}$$

$c^T = (1, -2, -1, -1, 4),\ b^T = (-8, 12),\ l^T = (0, 0, 0, 0, 0),\ u^T = (6, 2, 10, 10, 3)$

Input: $B = \{1, 2\},\ N = \{3, 4, 5\},\ A_B = \begin{pmatrix} 0 & -5 \\ 1 & 4 \end{pmatrix},\ N^+ = N$, d.h. $x_3 = x_4 = x_5 = 0$, $A_b x_B = b \Rightarrow$ $x_B^T = (\frac{28}{5}, \frac{8}{5})$ zulässig!

1. Iteration:

1. Schritt: $\pi^T A_B = c_B^T \ \Leftrightarrow\ (\pi_1, \pi_2)\begin{pmatrix} 0 & -5 \\ 1 & 4 \end{pmatrix} = (1, -2) \Rightarrow\ \pi = (\frac{6}{5}, 1)$

2. Schritt: $\mu_N^T = c_N^T - \pi^T A_N \ \Leftrightarrow\ \mu_j = c_j - \pi^T A_j,\ j \in N$

$$\begin{aligned} \mu_3 &= -1 - (\tfrac{6}{5}, 1)\tbinom{-2}{1} = \tfrac{2}{5} \\ \mu_4 &= -1 - (\tfrac{6}{5}, 1)\tbinom{4}{-3} = -\tfrac{9}{5} \qquad \Rightarrow\ x_4 \textit{ entering} \\ \mu_5 &= 4 - (\tfrac{6}{5}, 1)\tbinom{-6}{5} = \tfrac{31}{5} \end{aligned}$$

4. Schritt: $A_B w = A_{j_l} \Rightarrow \begin{pmatrix} 0 & -5 \\ 1 & 4 \end{pmatrix} \binom{w_1}{w_2} = \binom{4}{-3} \Rightarrow w^T = (\frac{1}{5}, -\frac{4}{5})$

Schrittweite $\hat{\theta}$: $x_B^{Neu} = x_B - \theta w$

$$0 \le \bar{x}_{B_1} = x_1 - \theta w_1 \le 6 \quad \Rightarrow \theta_1 = 28$$
$$0 \le \bar{x}_{B2} = x_2 - \theta w_2 \le 2 \quad \Rightarrow \theta_2 = \tfrac{1}{2}$$

$\hat{\theta} = \min\{\theta_1, \theta_2, u_4 - l_4\} = \{28, \frac{1}{2}, 10\} = \frac{1}{2} \quad \Rightarrow x_2$ *leaving*

5. Schritt: Update

$$x^{Neu} = x + \hat{\theta}\Delta x = x + \hat{\theta}\begin{pmatrix} -w \\ e_p \end{pmatrix}$$

$B \leftarrow \{1,4\},\ N \leftarrow \{3,2,5\},\ N^+ = \{3,5\},\ N^- = \{2\}$

$$x^{Neu} = x + \hat{\theta}\Delta x = \begin{pmatrix} \frac{28}{5} \\ \frac{8}{5} \\ 0 \\ 0 \\ 0 \end{pmatrix} + 1/2 \begin{pmatrix} -\frac{1}{5} \\ \frac{4}{5} \\ 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} \frac{11}{2} \\ 2 \\ 0 \\ \frac{1}{2} \\ 0 \end{pmatrix}$$

2. Iteration:

1. Schritt: $\pi^T A_B = c_B^T \Leftrightarrow (\pi_1, \pi_2)\begin{pmatrix} 0 & 4 \\ 1 & -3 \end{pmatrix} = (1, -1) \Rightarrow \pi = (\frac{1}{2}, 1)$

2. Schritt:

$$\mu_3 = -1 - (\tfrac{1}{2}, 1)\tbinom{-2}{1} = -1 \quad Kandidat$$
$$\mu_2 = -2 - (\tfrac{1}{2}, 1)\tbinom{-5}{4} = -\tfrac{7}{2} \quad ok$$
$$\mu_5 = 4 - (\tfrac{1}{2}, 1)\tbinom{-6}{5} = 2 \quad ok$$

4. Schritt: $A_B w = A_{j_l} \Rightarrow \begin{pmatrix} 0 & 4 \\ 1 & -3 \end{pmatrix} \binom{w_1}{w_2} = \binom{-2}{1} \Rightarrow w^T = (-\frac{1}{2}, -\frac{1}{2})$

Schrittweite $\hat{\theta}$: $x_B^{Neu} = x_B - \theta w$

$$0 \le x_1^{Neu} = \tfrac{11}{2} - \theta_1(-\tfrac{1}{2}) = \tfrac{11}{2} + \theta_1 \tfrac{1}{2} \le 6 \quad \Rightarrow \theta_1 = 1$$
$$0 \le x_4^{Neu} = \tfrac{1}{2} - \theta_4(-\tfrac{1}{2}) = \tfrac{1}{2} + \theta_4 \tfrac{1}{2} \le 10 \quad \Rightarrow \theta_4 = 19$$

$\hat{\theta} = \min\{\theta_1, \theta_4, u_3 - l_3\} = \{1, 19, 10\} = 1$

5. Schritt: Update

$$x^{Neu} = x + \hat{\theta}\Delta x = \begin{pmatrix} \frac{11}{5} \\ 2 \\ 0 \\ \frac{1}{2} \\ 0 \end{pmatrix} + 1 \begin{pmatrix} \frac{0}{5} \\ 0 \\ 0 \\ \frac{1}{2} \\ 0 \end{pmatrix} = \begin{pmatrix} 6 \\ 2 \\ 1 \\ 1 \\ 0 \end{pmatrix}$$

$B \leftarrow \{3,4\},\ N \leftarrow \{1,2,5\},\ N^+ = \{5\},\ N^- = \{1,2\},\ x_5^{Neu} = 0, x_1^{Neu} = 6, x_2^{Neu} = 2$

3. Iteration:

1. Schritt: $(\pi_1, \pi_2)\begin{pmatrix} -2 & 4 \\ 1 & -3 \end{pmatrix} = (-1, -1) \Rightarrow \pi = (2, 3)$

2. Schritt:

$$\begin{aligned}\mu_1 &= -1 - (2,3)\tbinom{0}{1} = -2 \quad \text{ok, weil } x_1 = u_1\\ \mu_2 &= -2 - (2,3)\tbinom{-5}{4} = -4 \quad \text{ok, weil } x_2 = u_2\\ \mu_5 &= 4 - (2,3)\tbinom{-6}{5} = 1 \quad \text{ok, weil } x_5 = l_5\end{aligned}$$

$\Rightarrow x^T = (6,2,1,1,0)$ ist optimal!

3.11 Rang-1-Korrektur

Die Rang-1-Korrektur ist ein Verfahren aus der Linearen Algebra. Es ist jedoch besonders interessant für das Simplex-Verfahren, da es dabei hilft, die Inverse einer Matrix explizit zu bestimmen. Das Ausgangsproblem ist von der Form $A_B w = y$ bzw. $A_B^T w = y$. Nun sei $PA_B = LU$ Daraus folgt:

$$PA_B w = L\underbrace{Uw}_{v} = Py =: b$$

Nun benötigt man zwei Schritte zum Berechnung von w:

1. Berechnung von v mit $Lv = b$
2. Berechnung von w mit $Uw = v$

Bei dieser Variante entsteht pro System ein Rechenaufwand von etwa m^2 Operationen bei einer Matrix $A_B \in M^{m\times m}$. Eine Zerlegung in dieser Weise ist also sehr aufwändig, was schließlich zu der Idee sogenannter „Updates“ führt:

Lemma 3.15 Seien $u, v \in \mathbb{R}^m$ Vektoren mit $1 + u^T v \neq 0$, so ist die Matrix $E := I + uv^T$ invertierbar und es ist

$$E^{-1} = I - \frac{uv^T}{1 + u^T v}\ .$$

Eine Matrix genau dieser Gestalt tritt num im Simplex-Algorithmus nach jedem Basisaustausch $j_{m+p} \leftrightarrow j_q$ auf:

$$A_B^{Neu} = A_B + (A_{\bullet j_{m+p}} - A_{\bullet j_q})e_q^T = A_B(I + (A_B^{-1}(A_{\bullet j_{m+p}} - A_B^{-1}A_{\bullet j_q})e_q^T) = A_B(\underbrace{I + (w - e_q)e_q^T}_{=:E})$$

Weil $1 + (w - e_q)^T e_q = 1 + w_q - 1 = w_q \neq 0$, da $\hat{\theta} = \frac{x_{j_q}}{w_q}$. Nach Lemma 3.15 ist somit nun

$$E^{-1} = I - \frac{(w - e_q)e_q^T}{w_q} = \begin{pmatrix} 1 & & -\frac{w_1}{w_q} & & \\ & \ddots & \vdots & & \\ & & 1 \quad -\frac{w_{q-1}}{w_q} & & \\ & & \frac{1}{w_q} & & \\ & & -\frac{w_{q+1}}{w_q} \quad 1 & & \\ & & \vdots & \ddots & \\ & & -\frac{w_m}{w_q} & & 1 \end{pmatrix}.$$

Nun braucht man nur noch einen Vektor w zu speichern und es ergibt sich

$$(A_B^{Neu})^{-1} = E^{-1}A_B^{-1} \ ,$$

$$(A_B^{Neu})^{-T} = A_B^{-T}E^{-T} \ .$$

Das Lösen der Gleichungssysteme geschieht nun folgendermaßen:

a) $A_B^{Neu}x = v$

i) Löse $A_B y = v$ (mit der LU-Zerlegung)

ii) Berechne $x = E^{-1}y$. Da $y = A_B^{-1}v$, ist $E^{-1}y = E^{-1}A_B^{-1}v = (A_B^{Neu})^{-1}v$ und es ergibt sich

$$x = y - \frac{(w - e_q)e_q^T y}{w_q} = y - (w - e_q)\frac{y_q}{w_q} = \begin{cases} y_i - \frac{w_i y_q}{w_q} & , \ i \neq q \\ y_q - \frac{(w_q - 1)y_q}{w_q} = \frac{y_q}{q_q} & , \ \text{sonst} \end{cases}$$

b) $(A_B^{Neu})^T x = v$

i) Berechne $y = E^{-T}v = v - \frac{e_q(w-e_q)^T v}{w_q} = \begin{cases} v_i & , \ i \neq q \\ v_q - \frac{(w-e_q)^T v}{w_q} = v_q - \frac{w^T v - v_q}{w_q} & , \ \text{sonst} \end{cases}$

ii) Löse $A_B^T x = y = E^{-T}v \quad \Rightarrow x = A_B^{-T}E^{-T}v = (A_B^{Neu})^{-T}v$

Implementierung:

- nach jedem Basistausch wird der Vektor w gespeichert. Diese nennt man die sogenannten „Eta-Files“.
- die obige Prozedur wird <u>rekursiv</u> angewendet

$$\begin{aligned} A_B &= A_B^{(0)}E_1 \cdot \ldots \cdot E_s \\ A_B^{-1} &= E_s^{-1} \cdot \ldots \cdot E_1^{-1}(A_B^{(0)})^{-1} \\ A_B^{-T} &= (A_B^{(0)})^{-T}E_1^{-T} \cdot \ldots \cdot E_s^{-T} \end{aligned}$$

- in regelmäßigen Abständen wird die Basismatrix neu zerlegt, um nicht zu viele „Eta-Schritte“ machen zu müssen und um numerische Stabilität zu gewährleisten

Definition 3.16 *Seien $\lambda_{max}, \lambda_{min}$ der maximale bzw. der minimale Eigenwert der Matrix A. Und sei ϱ der Spektralradius von A.*

$$cond_2(A) := \|A\|_2\|A^{-1}\|_2 = \varrho(A)\varrho(A^{-1}) \geq \frac{|\lambda_{max}|}{|\lambda_{min}|}$$

Oft wird der Begriff $cond_2$ auch durch κ_2 abgekürzt. Der Index verrät jeweils, welche Norm betrachtet wird.

Lemma 3.17 $cond_2(I + uv^T) \geq max\ (1 + u^T v,\ \frac{1}{1+u^T v})$

Beweis Eigenwerte und Eigenvektoren von $I + uv^T$

a) u ist Eigenvektor mit dem Eigenwert $1 + u^T v$, weil $(I + uv^T)u = u + u(v^T u) = (1 + u^T v)u$

b) Für alle $x \perp v$ mit dem Eigenwert 1 gilt: $(I + uv^T)x = x + u\underbrace{v^T x}_{=0} = x$ ■

Korollar 3.18

$$cond_2(E^{-1}) = cond_2\left(I - \frac{(w - e_q)e_q^T}{w_q}\right) = cond_2(E) = cond_2(I + (w - e_q)e_q^T) \geq \max\{w_q, \frac{1}{w_q}\}$$

Bemerkungen

- Bei den obigen Algorithmen kann ein *„Fill-in“* auftreten, das heißt das vorherige Nullelemente verschwinden.
- Der *„Fill-in“* hängt von der Anzahl der Elemente, die ungleich Null sind, in der Pivotzeile und der Pivotspalte ab.
- Es gibt allerdings Regeln, mit denen man ein „Fill-in“ vermeiden kann. Eine davon ist z.B. die Markowitz-Regel:

  Man wählt ein $|a_{ij}| \neq 0$, sodass $(p_i - 1)(q_j - 1)$ minimal ist. p_i ist hierbei die Anzahl der Nulllemente in der Zeile i und q_j die Anzahl der Nullelemente in der Spalte j. Wegen der numerischen Instabilität hierbei, kann man diese Regel noch modifizieren: Man bestimmt zuerst $\max |a_{ij}| =: \alpha$ und wählt danach $|a_{ij}| \leq 0,01\alpha$, sodass $(p_i - 1)(q_j - 1)$ minimal wird.

4 Dualitätstheorie

Wiederholung: Ist x eine zulässige Basislösung mit B und N und definiert man sich π und μ mit $c_B^T - \pi^T A_B = 0$ und $c_N^T - \pi^T A_N = \mu^T$, bzw.

$$c^T = (\pi^T, \mu^T) \begin{pmatrix} A_B & A_N \\ 0 & I \end{pmatrix} = \begin{pmatrix} \pi^T A_B \\ \pi^T A_N + \mu^T \end{pmatrix}^T ,$$

so gilt:

1. Sei x nicht degeneriert und optimal, dann ist $\mu \geq 0$. (notwendige Bedingung)
2. Ist $\mu \geq 0$, dann ist x optimal. (hinreichende Bedingung)

Der Beweis erfolgte durch den Darstellungssatz ($y = x + \sum_{i=m+1}^{n} y_i \eta^i$).

Bemerkung Ist x degeneriert und optimal, so existiert eine Basis, sodass $\mu \geq 0$.

Das **Ziel** der Dualitätstheorie ist es nun für ein lineares Programm der Form $\{\min c^T x,\ Ax = b,\ x \geq 0\}$ eine untere Schranke für $c^T x$ zu finden und auch somit zu überprüfen, ob ein x optimal ist. Die **Idee** hierfür ist das Bilden einer Linearkombination der Nebenbedingungen in der Art

$$y^T A + \mu^T I = y^T A + \mu^T$$

mit $y \in \mathbb{R}^m$ beliebig und $\mathbb{R}^n \ni \mu \geq 0$, sodass

$$y^T A + \mu^T I = c^T \ .$$

Es folgt:

$$c^T x = (y^T A + \mu^T)x = y^T Ax + \mu^T x = y^T b + \mu^T x \geq y^T b \ .$$

Somit ist $y^T b$ eine untere Schranke und y und μ erfüllen $y^T A + \mu^T = c^T$ mit $\mu \geq 0$. Die bestmögliche solcher Schranken erhält man nun über das Lösen des folgenden sogenannten **dualen LP's**:

$$\begin{array}{c} \max_y b^T y \\ A^T y + \mu = c \quad \Leftrightarrow \quad A^T y \leq c \\ \mu \geq 0 \end{array}$$

Beispiel

$$\begin{array}{ll} \max x_1 + x_2 & x_1 - x_2 \leq 0 \\ & -x_1 + x_2 \leq -1 \end{array}$$

$$\text{Standardform :} \quad \begin{array}{ll} \min -x_1 - x_2 & x_1 - x_2 \leq 0 \quad (i) \\ & -x_1 + x_2 \leq -1 \quad (ii) \end{array}$$

Die Addition der beiden Gleichungen (i) und (ii) führt zu $0 \leq -1$. Somit ist das primale Problem unzulässig. Um das primale LP nun in die duale Form umzuwandeln definiert man $x_i := x_i^+ - x_i^-$ mit $x_i^+, x_i^- \geq 0,\ i \in \{1, 2\}$. Außerdem ist die Einführung von zwei Schlupfvariablen notwendig. Insgesamt ergibt sich:

$c^T = (-1, 1, -1, 1, 0, 0)\ , \quad b^T = (0, -1)$

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

$\max b^T y = \ \max -y_2 \Leftrightarrow \min y_2$

$$A^T y + \mu = \begin{pmatrix} 1 & -1 \\ -1 & 1 \\ -1 & 1 \\ 1 & -1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \leq \begin{pmatrix} -1 \\ 1 \\ -1 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

$$\left.\begin{array}{l} y_1 - y_2 \leq -1 \\ -y_1 + y_2 \leq 1 \quad \Rightarrow y_1 - y_2 \geq -1 \end{array}\right\} \quad \Rightarrow y_1 - y_2 = -1 \quad (*)$$

$$\left.\begin{array}{l} -y_1 + y_2 \leq -1 \\ y_1 - y_2 \leq 1 \quad \Rightarrow -y_1 + y_2 \geq -1 \end{array}\right\} \quad \Rightarrow -y_1 + y_2 = -1 \quad (**)$$

Die Addition der beiden Gleichungen (*) und (**) führt zu $0 = -2$. Somit ist auch das duale Problem unzulässig. Das duale Programm hätte dann die folgende Form:

$$\min\ y_2 \qquad \begin{array}{l} y_1 - y_2 = -1 \\ -y_1 + y_2 = -1 \end{array} \qquad y_1, y_2 \leq 0$$

Lemma 4.1 Das duale Problem zu (D) ist wieder (P).

Beweis $\max\ b^T y \rightarrow \min\ -b^T y$. y hat keine untere Schranke, deshalb ist $y = y^+ - y^-$, $y^+, y^- \geq 0$ $\Rightarrow -b^T y = (-b^T, b^T)\binom{y^+}{y^-}$.

$$A^T y = A^T y^+ - A^T y^- = (A^T, -A^T)\begin{pmatrix} y^+ \\ y^- \end{pmatrix} \leq c$$

$$\text{oder} \quad \underbrace{(A^T, -A^T, I)}_{=:\bar{A}} \begin{pmatrix} y^+ \\ y^- \\ \gamma \end{pmatrix} = c =: \bar{b}\ , \quad \bar{c}^T := (-b^T, b^T, 0)$$

$$\text{duales Problem :} \qquad \max\ \bar{b}^T \xi = c^T \xi\ , \quad \bar{A}^T \xi \leq \bar{c}$$

Aus der Ungleichung folgt nun $A\xi \leq -b\ \wedge\ -A\xi \leq b\ \Rightarrow\ A\xi = -b$. Weiter folgt $\xi \leq 0$. Schließlich setzt man $x := -\xi$ und man ist wieder bei dem primale Problem angelangt. ■

Lemma 4.2 *(schwache Dualität)* Sei x primal zulässig, d.h. $Ax = b,\ x \geq 0$, und y dual zulässig, d.h. $A^T y \leq$ c, so ist

$$b^T y \leq c^T x\ .$$

Beweis Da $b^T = (Ax)^T$ folgt $b^T y - c^T x = x^T A^T y - x^T c = \underbrace{x^T}_{\geq 0}(\underbrace{A^T y - c}_{\leq 0}) \leq 0.$ ■

wichtige Bemerkung Der Wert der Zielfunktion im primalen (dualen) Problem ist obere (untere) Schranke für alle zulässigen dualen (primalen) Lösungen. Dies gilt insbesondere auch für die optimale Lösung.

Definition 4.3 *Sei*

$$\Delta := c^T x - b^T y = (c - A^T y)^T x = \gamma^T x \ .$$

Δ *heißt die* **Dualitätslücke** *(duality gab).*

Korollar Seien x primal zulässig und y dual zulässig sowie $\Delta = 0$, so sind x und y optimal.

Beweis $\Delta = 0 \Leftrightarrow c^T x = b^T y$. x und y nehmen demnach in der jeweiligen Zielfunktion den kleinstmöglichen bzw. größtmöglichen Wert an. ■

Satz 4.4 *(Dualitätstheorem der LP)*

(i) Hat (P) eine optimale Lösung, dann hat (D) eine optimale Lösung und es gilt

$$\min_x c^T x = \max_y b^T y \ .$$

(ii) Hat ein Problem eine unbeschränkte Zielfunktion, so hat das andere Problem eine unzulässige Lösung.

Beweis

(i) Sei x eine optimale, zulässige Lösung, dann existiert eine Basis mit $c_B^T - \pi^T A_B = 0$ und $c_N^T - \pi^T A_N = \mu_N^T \geq 0$. π ist dual zulässig (d.h. $A^T \pi \leq c$) und für die Dualitätslücke gilt:

$$\Delta = (c^T - \pi^T A)x = \underbrace{(c_B^T - \pi^T A_B)}_{=0} \cdot x_B + (c_N^T - \pi^T A_N) \cdot \underbrace{x_n}_{=0} = 0 \ .$$

Somit ist π dual optimal.

(ii) Ist die Negation von Lemma 4.2 ■

Bemerkungen

1. Der Multiplikatorenvektor π entsprechend zur primalen Optimallösung ist dual optimal. Das Simplex-Verfahren startet mit x primal zulässig, aber μ nicht ≥ 0, somit ist π nicht dual zulässig. Die Dualitätslücke Δ ist jedoch 0. Die Iteration läuft daher bis zur dualen Zulässigkeit von π.

2. Wenn (P) oder (D) unzulässig sind, folgt nicht, dass (D) oder (P) unbeschränkt ist. Beide können auch unzulässig sein.

		dual		
		optimal	unzulässig	unbeschränkt
	optimal	+	−	−
primal	unzulässig	−	+	+
	unbeschränkt	−	+	−

Satz 4.5 *(Komplementärer Schlupf)* Seien x, y primal bzw. dual zulässig. Notwendig und hinreichend für die Optimalität ist dann

$$(*) \quad (c^T - y^T A)x = 0 \quad \wedge \quad y^T(Ax - b) = 0 \ .$$

Beweis $\gamma = c - A^T y \geq 0$ und $x \geq 0$, dann $\gamma^T x = c^T x - b^T y \geq 0$. Falls $(*)$ gilt, so ist die Dualitätslücke geschlossen, wonach x und y optimal sind. Umgekehrt ist falls x, y optimal sind $\Delta = 0$ und es gelten die Bedingungen in $(*)$. ■

Bemerkung Für jedes beliebige primale LP existiert ein duales LP, sodass die Dualitätstheorie auf das primal-dual Paar anwendbar ist.

Beispiel Gegeben sei folgendes **primales LP**:

$$\min c^T x + d^T y \qquad \begin{matrix} Ax + By \leq a \\ Cx + Dy = b \end{matrix} \qquad x \geq 0$$

Sei nun z eine Schlupfvariable und $y = y^+ - y^-$, folgt die Standardform:

$$\begin{aligned} \min c^T x + d^T y^+ - d^T y^- + 0z \\ Ax + By^+ - By^- + Iz = a \\ Cx + Dy^+ - Dy^- + 0z = b \\ x, y^+, y^- \geq 0 \end{aligned}$$

Das zugehörige duale LP ist:

$$\begin{aligned} \max a^T v + b^T w \\ A^T v + C^T w \leq c \\ B^T v + D^T w \leq d \\ -B^T v - D^T w \leq -d \\ v \leq 0 \end{aligned}$$

Die beiden letzten Ungleichungen ergänzen sich zu einer Gleichung, sodass sich entgültig das folgenden **duale LP** ergibt:

$$\max a^T v + b^T w \qquad \begin{matrix} A^T v + C^T w \leq c \\ B^T v + D^T w = d \end{matrix} \qquad v \leq 0$$

Einige der möglichen Fälle beim Umformen von einem primalen in ein duales Programm sind nun allgemein in der folgenden Tabelle aufgelistet:

Primal	**Dual**
Gleichung/Ungleichung	Variable
Ungleichung	nicht-positive Variable
Gleichung	freie Variable
nicht negative Variablen	Ungleichung
freie Variablen	Gleichung
$\min c^T x, \quad Ax \leq b, \quad x \geq 0$	$\max b^T y, \quad y^T A \leq c^T, \quad y \leq 0$
$\min c^T x, \quad Ax \geq b, \quad x \geq 0$	$\max b^T y, \quad y^T A \leq c^T, \quad y \geq 0$

Satz 4.6 (Starker Komplementärer Schlupf) Gebegeben seien ein primales Problem $\{\min c^T x,\ Ax = b,\ x \geq 0\}$ und das duale Problem $\{\max b^T y,\ A^T y \leq c\}$. Besitzen beide zulässige Lösungen, so existieren optimale Lösungen x^* und y^* mit

$$\begin{aligned} x_j^* > 0 \quad &\Leftrightarrow \quad c_j - (A_{\bullet j})^T y^* = 0 \\ x_j^* = 0 \quad &\Leftrightarrow \quad c_j - (A_{\bullet j})^T y^* > 0 \end{aligned}$$

Korollar (Kuhn-Tucker-Bedingungen) x ist genau dann eine optimale Lösung des primalen Programms $\{\min c^T x,\ Ax = b,\ x \geq 0\}$, wenn ein y und ein γ existieren mit:

1. $Ax = b,\ x \geq 0$ (primale Zulässigkeit)
2. $A^T y + \gamma = c,\ \gamma \geq 0$ (duale Zulässigkeit, Stationarität)
3. $\gamma^T x = 0$, d.h. $\gamma_j = 0 \ \vee \ x_j = 0 \quad \forall j$ (Komplementarität)

Satz 4.7 (Lemma von Farkas) Es gilt

$$Ax = b,\ x \geq 0 \text{ lösbar} \quad \Leftrightarrow \quad A^T y \leq 0,\ b^T y > 0 \text{ unlösbar} .$$

Beweis Sei

$$(P) \quad \begin{array}{c} \min 0^T x \\ Ax = b \\ x \geq 0 \end{array} \qquad \text{und} \qquad (D) \quad \begin{array}{c} \max b^T y \\ A^T y \leq 0 \end{array} .$$

„⇒“ (D) ist zulässig, da mit $y = 0$ gilt $A^T y \leq 0$. Habe nun (P) eine primal zulässige Lösung, so besitzt das duale Programm eine optimale Lösung mit

$$b^T y \leq 0^T \cdot x = 0 .$$

Demnach besitzt (D) keine Lösung.

„⇐“ (2) habe eine Lösung, d.h. $\exists\ y$ mit $A^T y \leq 0$ und $b^T y > 0$. Somit ist $ky\ \forall\ k > 0$ dual zulässig, da $kA^T y \leq 0$, woraus jedoch folgt

$$kb^T y \overset{k \to \infty}{\longrightarrow} +\infty .$$

(D) ist somit unbeschränkt, also (P) unzulässig. Demnach besitzt (P) keine Lösung. ■

Bemerkung Dieser Satz von Julius Farkas aus dem Jahre 1902 ist einer der fundamentalsten in der Linearen Optimierung und der Spieltheorie. Er bedeutet nichts anderes, als dass von den beiden System (1) und (2) stets genau eins lösbar ist. Diese Aussage lässt sich auch geometrisch interpretieren: Zwei konvexe Polyeder P und Q sind genau dann durch eine Hyperebene trennbar, wenn ihr Durchschnitt $P \cap Q$ leer ist.

Folgerungen

(i) $Ax \leq b$ unlösbar $\Leftrightarrow$ $A^T y = 0,\ y \leq 0,\ b^T y > 0$ lösbar

Beweis $(A, -A, I) \begin{pmatrix} x^+ \\ x^- \\ z \end{pmatrix} = b,\ x^+, x^-, z \geq 0$ lösbar $\overset{\text{Farkas}}{\Leftrightarrow}$ $\begin{pmatrix} A^T \\ -A^T \\ I \end{pmatrix} y \leq 0,\ b^T y > 0$ lösbar .

(ii) Allgemeiner Fall:

$$\begin{array}{l} Ax + By \leq a \\ Cx + Dy = b \\ \quad x \geq 0 \end{array} \text{ unlösbar } \Leftrightarrow \begin{array}{c} A^T u + C^T v \leq 0 \\ B^T u + D^T v = 0 \\ u \leq 0 \\ a^T u + b^T v < 0 \end{array} \text{ lösbar}$$

4.1 Ökonomische Interpretation der Dualität

Seien wieder

$$(P) \quad \begin{array}{c} \min\ c^T x \\ Ax = b \\ x \geq 0 \end{array} \quad , \quad (D) \quad \begin{array}{c} \max\ b^T y \\ A^T y \leq c \end{array} \ .$$

Lemma 4.8 Sei x^* mit Basis B und Nichtbasis N eine nicht degenerierte optimale Basislösung und Δb eine kleine Störung von b. Sei weiter y^* eine optimale Lösung von (D), so ist $(x^*)^{neu}$ mit

$$c^T x^* \to c^T (x^*)^{neu} = c^T x^* + \Delta b^T y^*$$

eine optimale Lösung für das gestörte LP $P(\Delta b) := \{\min c^T x,\ Ax = b + \Delta b,\ x \geq 0\}$.

Beweis Da x^* nicht degeneriert, ist

$$x^* = \begin{pmatrix} x_B^* \\ x_N^* \end{pmatrix} = \begin{pmatrix} A_B^{-1} b \\ 0 \end{pmatrix} \quad \text{mit} \quad x_B^* = A_B^{-1} b > 0 \ .$$

Für „genügend“ kleine Δb ist somit

$$(x^*)^{neu} = \begin{pmatrix} A_B^{-1}(b+\Delta b) \\ 0 \end{pmatrix} = \begin{pmatrix} x_B^* + A_B^{-1}\Delta b) \\ 0 \end{pmatrix} = x^* + \begin{pmatrix} A_B^{-1}\Delta b \\ 0 \end{pmatrix} \geq 0$$

zulässig. Außerdem bleibt y^* auch für das gestörte System dual zulässig und es gilt

$$c^T(x^*)^{neu} - (b+\Delta b)^T y^* = c^T x^* + \underbrace{c_B^T A_B^{-1}}_{=(y^*)^T}\Delta b - b^T y^* - \Delta b^T y^* = 0 \ .$$

$(x^*)^{neu}$ ist demnach primal optimal in $P(\Delta b)$. ■

Korrolar Eine nichtdegenerierte Lösung x^* $(= x^*(b))$ hängt (lokal) differenzierbar von b ab und es gilt:

$$\frac{dx^*(b)}{db}\Delta b = \begin{pmatrix} A_B^{-1} \\ 0 \end{pmatrix}\Delta b \quad , \quad \frac{d(c^T x^*)}{db}\Delta b = (\pi^*)^T \Delta b$$

Der Wert π_i^* sagt aus, welchen Preis man bezahlt (bzw. erhält), wenn man die Ressource b_i $(= a_i^T x)$ um eine Einheit erhöht.

Bemerkung Die Multiplikatoren π heißen **Schattenpreise** (shadow prices).

4.2 Das duale Simplex - Verfahren

Im Folgenden betrachtet man wieder $\max b^T y$ mit $A^T y \leq c$. Mit Basis B und Nichtbasis N im primalen System ergibt sich

$$\begin{array}{ccc} & \min\ c^T x & \\ (P) \quad & (A_B, A_N)\binom{x_B}{x_N} = b & \Rightarrow \quad (D) \quad A_B^T y = c_B \\ & x_B \geq 0,\ x_N = 0 & \qquad\quad A_N^T y \leq c_N \end{array} \qquad \begin{array}{c} \max\ b^T y \\ \\ \\ \end{array}$$

Definition 4.9 *Primale Zulässigkeit: x ist eine zulässige Basislösung von (P), wenn*

$$x_N = 0,\ x_B = A_B^{-1} b \geq 0 \ .$$

Duale Zulässigkeit: π ist eine zulässig Basislösung von (D), wenn

$$A_B^T \pi = c_B,\ A_N^T \pi \leq c_N \ .$$

Satz 4.10

(a) π ist genau dann eine Ecke von $P_D = \{y | A^T y \leq c\}$, wenn π eine dual zulässige Basislösung ist.

(b) π ist optimal, falls $x_B = A_B^{-1} b \geq 0$ ist.

Beweis (a) Ähnlich wie im primalen Fall: $\pi \in \mathbb{R}^m$ wird durch m „aktive Bedingungen“ $A_B^T = c_B$ festgelegt.

(b) folgt aus der Dualitätstheorie. ■

Duales Simplexverfahren

Im Folgenden wird nun der Algorithmus für das duale Simplexverfahren formuliert. Ausgegangen wird hierbei von einer Matrix $A \in \mathbb{R}^{m \times n}$ mit $rang\ A = m$ und den beiden bekannten Problemen:

$$(P) \quad \begin{array}{c} \min\ c^T x \\ Ax = b \\ x \geq 0 \end{array} \quad , \quad (D) \quad \begin{array}{c} \max\ b^T y \\ A^T y \leq c \end{array} \quad \Leftrightarrow \quad \begin{array}{r} \max\ b^T y \\ A^T y + z = c \\ z \geq 0 \end{array}$$

Gegeben sei nun eine Basis $B = \{B_1, \ldots, B_m\} \subset \{1, \ldots, n\}$, sodass A_B regulär ist. $x = \binom{x_B}{x_N}$, $x_N = 0$, $z_N = c_N - A_N^T y$. Der Vektor $y := A_B^{-T} c_B$ ist nun eine duale Basislösung. Ist $z_N \geq 0$, so ist y eine zulässige duale Basislösung. Sei nun y zulässig und w eine weitere zulässige duale Lösung, so gilt:

$$y = A_B^{-T} c_B,\ c_N - A_N^T y \geq 0 \quad , \quad w \leq A_B^{-T} c_B,\ c_N - A_N^T w \geq 0$$

$$b^T y - b^T w = b^T (y - w) = x_B^T A_B^T (y - w) = x_B^T \underbrace{(c_B - A_B^T w)}_{\geq 0}$$

Ist nun $x_B \geq 0$, so ist $b^T y \geq b^T w\ \forall w$ dual zulässig $\Rightarrow$ y optimal.

Algorithmus

(1) Berechne x_B aus $x_B = A_B^{-1} b$.

(2) Prüfe Optimalität: Ist $x_B \geq 0$, **Stop!** y ist dual optimal und somit ist $x = \binom{x_B}{0}$ die primal optimale Lösung. Ansonsten setze den Algorithmus mit dem nächsten Schritt fort.

(3) Bestimme den Kandidatenindex B_p, der die Basis verlässt („leaving index“), mit $x_{B_p} < 0$.

(4) Berechne die Korrekturen u von y und v von z. Man berechnet u aus $A_B^T u = -e_p$, setzt $v_B = e_p$ und berechnet dann v_N aus $v_N = -A_N^T u$. Somit gilt nun

$$b^T y^{neu} - b^T y = -\sigma x_{B_p} \geq 0\ .$$

(5) Nun wird ein z bestimmt, sodass $z^{neu} = z + \sigma v \geq 0$. Falls $v_N \geq 0$ stoppt der Algorithmus, denn (D) ist unbeschränkt und (P) somit unzulässig. Ansonsten bestimme mit $N_q \in N$ den „entering

index“ mit $\sigma := \sigma_{N_q} = \min\{\sigma_j, j \in N\}$, wobei

$$\sigma_j := \begin{cases} -\frac{z_j}{v_j} & , \ v_j < 0 \\ \infty & , \ \text{sonst} \end{cases} .$$

(6) Als letzter Schritt erfolgt dann wie gewohnt das Basis-Update:

$$B^{neu} = (B \backslash \{B_p\}) \cup \{N_q\} \quad , \quad N^{neu} = (N \backslash \{N_q\}) \cup \{B_q\}$$

$$y^{neu} = y + \sigma u \quad , \quad z_N^{neu} = z_N + \sigma v_N \quad , \quad z_B^{neu} = e_p \sigma$$

Es folgen nun die Nebenrechnungen zu den Schritten vier und fünf:

(NR4) $y^{neu} = y + \sigma u$ und $z^{neu} = z + \sigma v$, wobei u und v die Korrekturen und σ die Schrittweite ist. Nun sollen y^{neu} und z^{neu} dualzulässig sein, d.h.

$$z^{neu} = c - A^T y^{neu} \geq 0 .$$

$$\Rightarrow \quad z^{neu} = c - A^T(y + \sigma u) = c - A^T y - \sigma A^T u = z - \sigma A^T u$$

$$\Rightarrow \quad \sigma v = -\sigma A^T u \quad \text{bzw.} \quad v = -A^T u \quad \Leftrightarrow \quad \begin{pmatrix} v_B \\ v_N \end{pmatrix} = \begin{pmatrix} -A_B^T u \\ -A_N^T u \end{pmatrix}$$

Wie sollte nun v_B gewählt werden? Mit der Idee $b^T y^{neu} \overset{!}{\geq} b^T y$ ergibt sich nun

$$b^T y^{neu} - b^T y = x_B^T A_B^T (y^{neu} - y) = \sigma x_B^T A_B^T u = -\sigma x_B^T v_B \overset{!}{\geq} 0 .$$

Setzt man nun $v_B := e_p$, so folgt $-\sigma x_B^T v_B \geq 0$, da $x_{B_p} \leq 0$ nach Wahl von p.

(NR5) Mit dem in Schritt 4 berechneten Korrekturvektor v für z, ergibt sich für $z_j^{neu} = z_j + \sigma v_j$ nun

$$z_j^{neu} = \begin{cases} z_j = 0 & , \ j \in B \backslash \{B_p\} \\ z_j + 1 \cdot \sigma = \sigma \geq 0 & , \ j = B_p \\ z_j + \sigma_j v_j \overset{!}{\geq} 0 & , \ j \in N \end{cases} .$$

Deshalb wählt man für $j \in N$ die σ_j mit

$$\sigma_j := \begin{cases} -\frac{z_j}{v_j} & , \ v_j < 0 \\ \infty & , \ \text{sonst} \end{cases} .$$

Beispiel Sei das LP

$$\begin{array}{c} \min 2x_1 + x_2 \\ -x_1 - x_2 \leq -\frac{1}{2} \\ -4x_1 - x_2 \leq -1 \\ x \geq 0 \end{array} \quad \Leftrightarrow \quad \begin{array}{c} \min 2x_1 + x_2 \\ -x_1 - x_2 + x_3 = -\frac{1}{2} \\ -4x_1 - x_2 + x_4 = -1 \\ x \geq 0 \end{array}$$

gegeben. Um den ersten Iterationsschritt zu beginnen setzt man nun $B = \{3, 4\}$ und $N = \{1, 2\}$. Somit ist $z_N^T = c_N^T - Ay^T A_N = c_N^T - (c_B^T A_B^{-1})A_N = (2,1) - (0,0)A_B^{-1}A_N = (2,1) \geq 0$ und somit B dual zulässig.

Iterationsschritt 1

(1) Zuerst wird das x_B berechnet und geprüft, ob es nicht vielleicht schon die Optimallösung ist.

$$x_B = I\begin{pmatrix} -\frac{1}{2} \\ -1 \end{pmatrix} = \begin{pmatrix} -\frac{1}{2} \\ -1 \end{pmatrix} = \begin{pmatrix} x_3 \\ x_4 \end{pmatrix} < 0 \quad \Rightarrow \quad \text{nicht optimal}$$

(3) Nun wähle ein B_j mit $x_{B_j} < 0$. Wähle $B_j = 4,\ j = 2$ mit $x_4 = -1 < 0$. Also ist $B_2 = 4$ der „leaving Index“.

(4) Berechne nun die Korrekturen u für y, v_N für z_N und v_B für z_B. $v_B := e_2 = \binom{0}{1}$ zu $\binom{z_3}{z_4}$. Berechne u aus $A_B^T u = -e_j \ \Rightarrow\ u = -\binom{0}{1}$ und schließlich

$$v_N = -A_N^T u = \begin{pmatrix} -1 & -4 \\ -1 & -1 \end{pmatrix}\begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} -4 \\ -1 \end{pmatrix} \text{ zu } \begin{pmatrix} z_1 \\ z_2 \end{pmatrix}.$$

(5) Berechne nun die Schrittweite σ:

$$z_1 = 2 \quad \Rightarrow \quad z_1^{neu} = z_1 + \sigma_1 v_1 = 2 + \sigma_1(-4) \overset{!}{\geq} 0 \quad \Rightarrow \quad \sigma_1 = \frac{1}{2}$$

$$z_2 = 1 \quad \Rightarrow \quad z_2^{neu} = z_2 + \sigma_2 v_2 = 1 + \sigma_2(-1) \overset{!}{\geq} 0 \quad \Rightarrow \quad \sigma_2 = 1$$

Nun ist $\sigma = \min\{\sigma_1, \sigma_2\} = \frac{1}{2}$. Also ist $N_1 = 1$ der „entering Index“.

(6) Basis-Update

$$N \leftarrow \{4,2\},\ B \leftarrow \{3,1\},\ z \leftarrow \begin{pmatrix} z_1^{neu} \\ \vdots \\ z_4^{neu} \end{pmatrix} = \begin{pmatrix} z_1 + \sigma v_1 \\ z_2 + \sigma v_2 \\ z_3 + \sigma v_3 \\ z_4 + \sigma v_4 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 + \frac{1}{2}(-1) \\ 0 \\ 0 + \frac{1}{2} \cdot 1 \end{pmatrix} = \begin{pmatrix} 0 \\ \frac{1}{2} \\ 0 \\ \frac{1}{2} \end{pmatrix}$$

Iterationsschritt 2 Im folgenden wird nun der Algorithmus wieder solange wiederholt, bis man entweder eine Optimallösung gefunden hat, oder festellen muss, dass das Problem unbeschränkt bzw. unzulässig ist.

(1) Berechne x_B aus

$$A_B x_B = b \Leftrightarrow \begin{pmatrix} 1 & -1 \\ 0 & -4 \end{pmatrix}\begin{pmatrix} x_3 \\ x_1 \end{pmatrix} = \begin{pmatrix} -\frac{1}{2} \\ -1 \end{pmatrix} \Rightarrow \begin{pmatrix} x_3 \\ x_1 \end{pmatrix} = \begin{pmatrix} -\frac{1}{4} \\ \frac{1}{4} \end{pmatrix} \not\geq 0 \Rightarrow \text{ nicht optimal}$$

(3) Wähle $B_1 = 3$, da $x_3 < 0$, als „leaving Index“

(4) $v_B = \binom{1}{0}$ zu $\binom{z_3}{z_1}$. Berechne u aus $\begin{pmatrix} 1 & 0 \\ -1 & -4 \end{pmatrix} u = -\binom{1}{0} \Rightarrow u = \binom{-1}{\frac{1}{4}}$.

$$v_N = -A_N^T u = -\begin{pmatrix} 0 & 1 \\ -1 & -1 \end{pmatrix}\begin{pmatrix} -1 \\ \frac{1}{4} \end{pmatrix} = \begin{pmatrix} -\frac{1}{4} \\ -\frac{3}{4} \end{pmatrix} \text{ zu } \begin{pmatrix} z_4 \\ z_2 \end{pmatrix}$$

(5)

$$z_4^{neu} = z_4 + \sigma_4 v_4 = \frac{1}{2} + \sigma_4(-\frac{1}{4}) = 0 \quad \Rightarrow \quad \sigma_4 = 2$$

$$z_2^{neu} = z_2 + \sigma_2 v_2 = \frac{1}{2} + \sigma_2(-\frac{3}{4}) = 0 \quad \Rightarrow \quad \sigma_2 = \frac{2}{3}$$

Somit ist $\sigma = \sigma_2 = \frac{2}{3}$, also $N_2 = 2$ der „entering index“.

(6) Basis-Update

$$N \leftarrow \{4,3\},\ B \leftarrow \{2,1\},\ z = \begin{pmatrix} 0 + \sigma \cdot 0 \\ \frac{1}{2} + \sigma \cdot (-\frac{3}{4}) \\ 0 + \sigma \cdot 1 \\ \frac{1}{2} + \sigma(-\frac{1}{4}) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \frac{2}{3} \\ \frac{1}{3} \end{pmatrix}$$

Iterationsschritt 3

$$A_B x_B = b \Leftrightarrow \begin{pmatrix} -1 & -1 \\ -1 & -4 \end{pmatrix}\begin{pmatrix} x_2 \\ x_1 \end{pmatrix} = \begin{pmatrix} -\frac{1}{2} \\ -1 \end{pmatrix} \Rightarrow \begin{pmatrix} x_2 \\ x_1 \end{pmatrix} = \begin{pmatrix} \frac{1}{3} \\ \frac{1}{6} \end{pmatrix} \geq 0 \Rightarrow\ x_B \text{ optimal}$$

Insgesamt ist somit $x^T = (\frac{1}{6}, \frac{1}{3}, 0, 0)$ die Optimallösung der Linearen Programms.

Bemerkungen

(i) Der Duale-Simplexalgorithmus ist für das Problem $\max b^T y,\ A^T y \leq c$ der gleiche wie für den primalen Simplexalgorithmus

(ii) Für ein LP ist der Ablauf des primalen und des dualen Simplexalgorithmus ganz verschieden.

(iii) Der duale Simplexalgorithmus ist auch in der gemischt-ganzzahligen Optimierung anwendbar.

4.3 Postoptimale Analyse

Die postoptimale Analyse beschäftigt sich mit den drei folgenden auftretenden Problemen: Zum einen sind die Daten des Linearen Programms (A,b,c) nur ungenau bekannt, aber veränderlich. Außerdem können eventuell zusätzliche Entscheidungsvariablen x_j sowie zusätzliche Restriktionen eingeführt werden.
Allgemein wird in den nun folgenden Verfahren von dem Standardproblem $\{\min\ c^T x,\ Ax = b,\ x \geq 0\}$ ausgegangen. Für das Problem ist die optimale Lösung x mit B, N bekannt.

4.3.1 Änderung von b

In dem obigen Standardproblem setzt man nun $b' = b + \Delta b$, also $Ax = b'$. Da allerdings gilt

$$\mu_N = c_N - A_N^T \pi = c_N - A_N^T A_B^{-T} c_B \geq 0 \ ,$$

bleibt das gestörte LP dual zulässig. Nun berechnet man x'_B aus

$$x'_B = A_B^{-1} b' = A_B^{-1}(b + \Delta b) = x_B + A_B^{-1}\Delta b = x_B + \Delta x_B \ .$$

Für x_N ändert sich hierbei nichts. Gilt nun $x'_B \geq 0$, so ist x' primal zulässig und es folgt aus der dualen Zulässigkeit, dass auch x' schon otimal für das gestörte LP ist und es gilt

$$c^T x' = c^T x + c^T \Delta x = c^T x + c_B^T \Delta x_B = c^T x + c_B^T A_B^{-1} \Delta b = c^T x + \pi^T \Delta b \ .$$

Ist dies jedoch nicht der Fall, so startet man den dualen Simplex-Algorithmus mit B, N für die Nachoptimierung bis man die primale Zulässigkeit erreicht hat. Für die Zielfunktion ergibt sich hierbei

4.3.2 Änderung der Zielfunktion

Ändert man die Zielfunktion in der Form $c' = c + \Delta c$, so hat dies keine Auswirkungen für $x \geq 0$. Deshalb bleibt hierbei die primale Zulässigkeit weiter erhalten. Nun betrachtet man den Vektor der reduzierten Kosten

$$\mu'_N = (c_N + \Delta c_N) - A_N^T A_B^{-T}(c_B - \Delta c_B) = \mu_N + \underbrace{\Delta c_N - A_N^T \Delta \pi}_{=:\Delta \mu_N} = \mu_N + \Delta \mu_N \ .$$

Ist nun $\mu'_N \geq 0$, so ist das LP auch dual zulässig, wonach dann x auch für das gestörte LP optimal ist. Es ergibt sich

$$(c')^T x = (c_N + \Delta c_N)^T x_N + (c_B + \Delta c_B)^T A_B^{-1} b = c^T x + \Delta c_N^T x_N + \Delta c_B^T A_B^{-1} b = c^T x + \Delta \pi^T b \ .$$

Ist jedoch $\mu'_N \not\geq 0$, so startet man für die Nachoptimierung den primalen Simplex-Algorithmus mit x, B, N, bis man auch die duale Zulässigkeit erreicht hat.

4.3.3 Änderung der Matrix A

(i) Nun betrachtet man eine Änderung in der j-ten Spalte der Matrix A. Es sei $A'_{\bullet j} = A_{\bullet j} + \Delta a$.

1. *Fall* Ist $j \in N$, so ist $x' = x$ weiterhin primal zulässig. Nun betrachtet man

$$\mu'_N = c_N - (A_N + \Delta a e_J^T)^T A_B^{-T} c_B = \mu_N - e_j \Delta a^T \pi \ .$$

Es ist also $\mu'_k = \mu \ \forall \ k \in N \backslash \{j\}$. Es ist somit ausreichend den Eintrag μ'_j zu überprüfen. Ist $\mu'_j \geq 0$, so folgt auch hier die duale Zulässigkeit von x und somit dessen Optimalität. Gilt jedoch $\mu'_j < 0$, so startet man wieder den primalen Simplex-Algorithmus, um das LP weiter zu optimieren.

2. *Fall* Ist jedoch $j \in B$, so wurde eine Änderung in der Basismatrix vorgenommen, was den so ziemlich schlimmst mögliche Fall darstellt. Hierbei kann es sogar passieren, dass die Regularität von A verloren gegangen ist. In dem Fall muss man für das LP eine neue Anlaufrechnung (Phase I) durchführen. Auf jeden Fall muss jedoch die Inverse A_B^{-1} neu bestimmt werden. Im Anschluss kann nun der primale Simplex-Algorithmus fortgeführt werden.

(ii) Bei der Änderung einer Zeile i sind die auftretenden Probleme fast identisch. Auch hier muss eine Entscheidung zwischen Basis und Nichtbasis getroffen werden. Bei einer Änderung in A_B muss in der Regel der primale Simplex-Algorithmus mit einer Phase-I Anlaufrechnung begonnen werden. Tritt allerdings eine Änderung in A_N auf, so bleibt die primale Zulässigkeit auf jeden Fall bestehen. Die duale Zulässigkeit könnte allerdings verloren gegangen sein. In diesem Falle kann man das gestörte LP mit dem primalen Simplex-Algorithmus nachoptimieren.

4.3.4 Hinzufügen von Variablen

Fügt man dem LP eine zusätzliche Variable x_{n+1} hinzu, so bleibt die Lösung $x' = \binom{x}{x_{n+1}}$ mit $x_{n+1} := 0$ auf jeden Fall primal zulässig. Es bleibt also zu überprüfen, ob die neue Komponente einen Einfluss auf die duale Zulässigkeit hat. Es ist

$$(\mu'_N)^T = (c'_N)^T - \pi^T A'_N = (\mu_N^T, c_{n+1} - \pi^T A_{\bullet n+1}) \ .$$

Ist also $\mu'_{n+1} = c_{n+1} - \pi^T A_{\bullet n+1} \geq 0$, so ist x' auch für das erweiterte LP eine Optimallösung. Ist dies nicht der Fall, so startet man zur Nachoptimierung den primalen Simplex-Algorithmus mit $x' = \binom{x}{0}$, $B' = B$ und $N' = N \cup \{n+1\}$.

4.3.5 Hinzufügen einer Restriktion

Nun wird das Standardproblem um eine Restriktion, d.h. die Matrix A und der Vektor b um eine Zeile bzw. eine Komponente erweitert. x^*, π^* und $\mu^* = c - A^T\pi^*$ seien hierbei für das Ausgangsproblem optimal. Als erweitertes LP erhält man

$$(P') \qquad \begin{array}{c} \min \ c^T x \\ Ax = b \\ A_{m+1,\bullet} \ x = b_{m+1} \\ x \geq 0 \end{array} \ .$$

Durch diese Erweiterung ist die Basismatrix A_B jedoch nicht mehr regulär. Man müsste somit eine neue Anlaufrechnung (Phase I) durchführen. Um dies zu umgehen erweitern wir auch den Vektor x um eine Komponente $x_{n+1} := 0$ und setzen $B' = B \cup \{n+1\}$, $N' = N$. Um nun auch die Matrix A' entsprechend anzupassen, setzt man $A'_{\bullet,n+1} := e_{m+1}$. Als neue Basismatrix erhält man somit

$$A'_B = \begin{pmatrix} A_B & 0 \\ A_{m+1,B} & 1 \end{pmatrix} \quad \text{mit} \quad \det A'_B \neq 0 \ .$$

Nun berechnet man π' für das erweitere System. Es ergibt sich

$$(\pi')^T A'_B = (c'_B)^T \quad \Leftrightarrow \quad \begin{matrix} \pi'_{m+1} = 0 \\ \pi^T A_B = c_B^T \end{matrix} \quad \Leftrightarrow \quad \begin{matrix} \pi'_{m+1} = 0 \\ \pi = \pi^* \end{matrix} \ .$$

Hieraus ergeben sich direkt die reduzierten Kosten

$$\mu'_N = c'_N - (A'_N)^T \pi' = c_N - \left(A_N^T, A_{m+1,\bullet}^T\right) \begin{pmatrix} \pi^* \\ 0 \end{pmatrix} = A_N^T \pi^* = \mu_N^* \ .$$

Somit bleibt auch für das erweiterte LP die duale Zulässigkeit erhalten.

a) Falls nun x^* die Bedingung $A_{m+1,\bullet}\ x^* = b_{m+1}$ erfüllt, so ist x^* primal zulässig in (P') und demnach schon optimal.

b) Ist jedoch x^* nicht zulässig, so startet man zur Nachoptimierung den dualen Simplex-Algorithmus mit $B',\ N',\ x' = \binom{x^*}{0}$ und A'.

5 Innere-Punkt-Verfahren

Innere-Punkte-Verfahren (engl. interior point method) sind in der Optimierung eine Klasse von Algorithmen zur Lösung von Optimierungsaufgaben. Ihr Hauptanwendungsgebiet sind lineare oder quadratische Programme. Im Vergleich zum Simplex-Verfahren zeichnen sich Innere-Punkte-Verfahren durch bessere theoretische Eigenschaften und schnellere Konvergenz für sehr große dünnbesetzte Probleme aus. Ein Nachteil ist, dass sie vergleichbar ungeeignet zum Lösen einer Serie von Optimierungsaufgaben sind. Um die möglichen Verbesserungen durch die Innere-Punkte-Verfahren er erforschen, sollen vorab noch einmal kurz die beiden bisher bekannten Algorithmen wiederholt werden:

Bei dem **primalen Simplex-Algorithmus** geht man von dem Standard LP $(P) = \{\min\ c^T x,\ Ax = b,\ x \geq 0\}$ aus. Sei hierbei x eine zulässige Basislösung, so sucht man duale Variablen π und μ mit

$$c^T x = c_B^T x_B = \underbrace{c_B^T A_B^{-1}}_{=\pi^T} b = b^T \pi \ .$$

Sind nun π und μ dual zulässig, so folgt hieraus bereits die Optimalität.

Der **duale Simplex-Algorithmus** hingegen besitzt das Ausgangsproblem $(D) = \{\max\ b^T\pi,\ A^T\pi + z = c,\ z \geq 0\}$. Seien nun π mit $z \geq 0$ eine dual zulässige Basislösung, so gilt für das zu der Basis B passende x nun wieder $b^T\pi = c^T x$. Falls x zudem noch zulässig ist, d.h. $x \geq 0$, so folgt die Optimalität.

Geometrisch bedeutet die Durchführung des Simplex-Algorithmus ein Springen von einer Ecke zur nächsten Ecke eines Polyeders. Jede Ecke ist ein Basislösung und wie bereits verifiziert liegt auch das Optimum in einer dieser Ecken. Im Gegensatz dazu versucht das „Innere-Punkt-Verfahren“ einen Pfad durch das Innere des Polyeders zum Optimum zu finden. Daher auch der Name.

Die Laufzeit des Simplex-Algorithmus ist, wie bereits erwähnt, nicht für alle Probleme optimal. Die Anzahl der Iterationen ist exponentiell zu $n,\ m$ und L, wobei L für die Anzahl der „Nichtnullstellen“ in $A \in \mathbb{R}^{m \times n}$ steht.

Das Ziel ist es nun, eine Methode mit polynomialer Laufzeit zu finden. Als Geburtsstunde der Inneren-Punkte-Verfahren, gilt gemeinhin die Arbeit von *Narendra Karmarkar* von 1984, die zum ersten Mal einen polynomialen potentiell praktisch einsetzbaren Algorithmus für lineare Probleme beschreibt und somit einen Aufwand von $\mathcal{O}(log\ n)$ besitzt.

5.1 Notation und theoretischer Hintergrund

Den oben erwähnten primalen und dualen Simplex-Algorithmen liegen die jeweiligen bekannten Linearen Programme (P) und (D) zu Grunde. Hat man eine primale Optimallösung x und eine duale Optimallösung y gefunden, so ist

- x primal zulässig, d.h. $Ax = b,\ x \geq 0$,

- y dual zulässig, d.h. $A^T y + z = c,\ z \geq 0$, und
- aus der Komplementarität folgt $z^T x = 0$, also $z_i x_i = 0\ \forall i = 1, ..., n$. Der Beweis hierfür ist recht einfach, da wegen der Optimalität von x und y die Dualitätslücke geschlossen ist und somit folgt

$$0 = c^T x - b^T y = c^T x - (Ax)^T y = x^T (c - A^T y) = x^T z \ .$$

Diese drei Argumente für eine Optimalität sind ein Spezialfall der *Karush-Kuhn-Tucker-Bedingungen* (KKT-Bedingungen) aus der nichtlinearen Optimierung mit

$$(NLP) \qquad \begin{array}{c} \min_x f(x) \\ g(x) = 0 \\ h(x) \geq 0 \end{array} \ ,$$

wobei hier $f, g, h \in \mathcal{C}^2,\ x \in \mathbb{R}^n,\ f : \mathbb{R}^n \to \mathbb{R},\ g : \mathbb{R}^n \to \mathbb{R}^m,\ h : \mathbb{R}^n \to \mathbb{R}^k$. In diesem Fall lautet die notwendige Bedingung 1. Ordnung:
Sei x^* optimal und regulär in (NLP), so existieren „Lagrange-Multiplikatoren“ $y \in \mathbb{R}^m$ und $z \in \mathbb{R}^k$, sodass

1. $g(x^*) = 0,\ h(x^*) \geq 0$
2. $\frac{\partial L}{\partial x}(x^*, y^*, z^*) = 0$, wobei $L = f(x) - y^T g(x) - z^T h(x)$ die Lagrange-Funktion
3. $(z^*)^T h(x^*) = 0$

Fügt man Punkt 2. schließlich noch die Bedindung $z \geq 0$ hinzu, so hat man den Spezialfall von oben.

Notation

$$e := \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} \in \mathbb{R}^n,\ X := \begin{pmatrix} x_1 & & 0 \\ & \ddots & \\ 0 & & x_n \end{pmatrix} \in \mathbb{R}^n,\ Z := \begin{pmatrix} z_1 & & 0 \\ & \ddots & \\ 0 & & z_n \end{pmatrix} \in \mathbb{R}^n$$

$$\Rightarrow \quad XZ = \begin{pmatrix} x_1 z_1 & & 0 \\ & \ddots & \\ 0 & & x_n z_n \end{pmatrix} \in \mathbb{R}^n$$

Mit dieser Notation erhält man schließlich:

$$\mathbf{KKT - LP}\ : \quad \begin{array}{c} Ax = b \\ A^T y + z = c \\ XZe = 0 \\ x, z \geq 0 \end{array}$$

5.2 Das Newton-Verfahren

Hier wird das Newton-Verfahren im $\mathbb{R}^n$ behandelt. Das Newtonsche Näherungsverfahren ist in der Mathematik das Standardverfahren zur numerischen Lösung von nichtlinearen Gleichungen und Gleichungssystemen. Im Falle einer Gleichung mit einer Variablen lassen sich zu einer gegebenen stetig differenzierbaren Funktion Näherungswerte zu Lösungen der Gleichung f(x) = 0, d.h. Näherungen der Nullstellen dieser Funktion finden.

Sei nun $F(w) = 0,\ F : D \subset \mathbb{R}^n \to \mathbb{R}^n,\ F \in \mathcal{C}^2$. Sei w^0 der Startwert der Iterationen. Die Iteration ist:

$$w^{i+1} = w + t_i \Delta w^i \qquad \text{mit } t_i \in]0, 1] \subset \mathbb{R}$$

Das Inkrement Δw^i ist hierbei aus der Taylorreihe von F um w^i. Somit ist

$$F(w) = F(w^i) + J(w^i)(w - w^i) \ + \ \text{Therme höherer Ordnung} .$$

Hierbei ist $J = F' \in \mathbb{R}^{n \times n}$ die sogenannte Jacobi-Matrix mit $J_{ij} = \frac{\partial f_i}{\partial x_j}$. Nun löst man die Gleichung

$$\Delta w^i = -J(w^i)^{-1} F(w^i) .$$

Dieses unendlich oft fortgesetzte Iterations-Verfahren konvergiert im günstigsten Fall mit quadratischer Konvergenzordnung

Newton-Methode:

$$F = F(x, y, z) = \begin{pmatrix} Ax - b \\ A^T y + z - c \\ XZe \end{pmatrix} = 0$$

x^0, y^0, z^0 sind hierbei die Startwerte. Für die Iteration folgt dann:

$$\begin{pmatrix} x^{i+1} \\ y^{i+1} \\ z^{i+1} \end{pmatrix} = \begin{pmatrix} x^i \\ y^i \\ z^i \end{pmatrix} + t^i \begin{pmatrix} \Delta x^i \\ \Delta y^i \\ \Delta z^i \end{pmatrix}$$

Der Vektor $(\Delta x^i, \Delta y^i, \Delta z^i)^T$ löst nun die Gleichung

$$\begin{pmatrix} A & 0 & 0 \\ 0 & A^T & I \\ Z^i & 0 & X^i \end{pmatrix} \begin{pmatrix} \Delta x^i \\ \Delta y^i \\ \Delta z^i \end{pmatrix} = - \begin{pmatrix} Ax^i - b \\ A^T y^i + z^i - c \\ X^i Z^i e \end{pmatrix} \overset{(*)}{=} \begin{pmatrix} 0 \\ 0 \\ X^i Z^i e \end{pmatrix} .$$

Hierbei gilt $(*)$ jedoch nur, wenn $x^i,\ y^i,\ z^i$ zulässig sind. Außerdem sind die Bedingungen $z, x \geq 0$ sehr wichtig, da sonst weitere Lösungen existieren, die nicht unbedingt zulässig sind. Pro Iteration gibt es nun jeweils zwei Schritte:

1. Lösen der linearen Gleichungssystem
2. Wahl der Schrittweite t

Beispiel 2:

$$\begin{array}{lll} & \min x_1 & \\ (P) & x_1 + x_2 = 1 & \\ & x \geq 0 & \end{array} \qquad \begin{array}{ll} & \max y \\ (D) & y \leq 1 \leftarrow \text{redundant} \\ & y \leq 0 \end{array}$$

Die Lösung für dieses LP ist $x^T = (0,1),\ y = 0$ und $z^T = (1,0)$. Ohne die Positivität von x und z gäbe es mit $x^T = (1,0),\ y = 1,\ z^T = (0,-1)$ noch eine weitere Lösung. An diesem Beispiel sieht man also, warum die Bedingungen $x, y \geq 0$ so wichtig sind.
Viele Innere-Punkt-Verfahren betrachten sogar nur Lösungen bzw. Punkte, die strikt im Inneren des Polyeders liegen, welche also die Bedingungen $x, y > 0$ erfüllen. Für diese Verfahren ergibt sich somit der entsprechende zuläsige Bereich

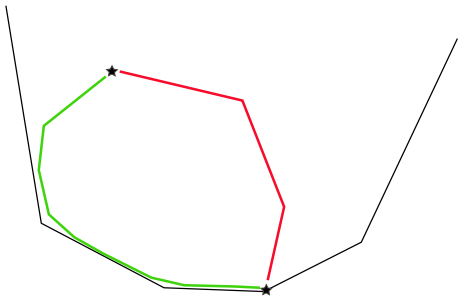
$$\mathfrak{F}^0 = \{(x,y,z) \mid Ax = b,\ A^T y + z = c,\ x, z > 0\}\ .$$

Korollar Ist A regulär und sind $x, y > 0$, so ist auch die Matrix

$$\begin{pmatrix} A & 0 & 0 \\ 0 & A^T & I \\ Z & 0 & X \end{pmatrix}$$

regulär.

5.3 Zentraler Pfad



Bei dem Standard-Newton-Verfahren macht man im Verlauf von jeder Iteration einen Schritt, der in Richtung des Randes des Polyeders $\mathfrak{F}^0$ geht. Hierbei entsteht das Problem, dass man somit immer nur eine sehr kleine Schrittweite $t << 1$ wählen kann und demnach entsprechend viele Schritte benötigt, bis das Verfahren gegen den optimalen Wert konvergiert.
Um dieses Problem zu umgehen, versucht man deshalb einen Weg durch das Innere des Polyeders zum Optimum zu finden, da in diesem Fall die Schrittweite enorm vergrößert werden könnte. Um einen solchen Weg durch das Innere des Polyeders zu charakterisieren, benötigen wir die folgende Definition.

Definition 5.1 *Der* **zentrale Pfad** *ist eine von $\tau > 0$ abhängige Kurve*

$$C = \{(x_\tau, y_\tau, z_\tau) \mid \tau > 0\}\ .$$

Die Punkte (x_τ, y_τ, z_τ) erfüllen hierbei die folgenden Bedingungen:

$$Ax - b = 0 \quad , \quad A^T y + z - c = 0 \quad , \quad x_i z_i = \tau\ \forall\ 1 \leq i \leq n \quad , \quad x, z > 0$$

Eigenschaften:

i) C ist eindeutig $\forall\ \tau > 0 \quad \Leftrightarrow \quad \mathfrak{F}^0 \neq \varnothing$

ii) Für $\tau \to 0$ konvergiert (wenn überhaupt) der Vektor (x_τ, y_τ, z_τ) gegen die optimale Lösung.

Statt $F(x, y, z) = 0$ löst man nun das LGS

$$F(x, y, z) = \begin{pmatrix} 0 \\ 0 \\ \tau e \end{pmatrix} \quad , \quad x, z > 0 \ .$$

Dieses modifizierte Verfahren bietet nun einige Vorteile. Zum einen sind nun X und Z immer strikt positiv. Außerdem werden nun die Produkte $x_i z_i$ gleichmäßig reduziert. Der größte Vorteil jedoch ist, dass man nun im Rahmen eines Iterationsschrittes eine größte Schrittweite t wählen kann.

Frage: Wie wählt man den optimalen Wert für τ?
Hierfür wählt man sich zuerst ein $\sigma \in [0, 1]$ als Zentrierungsparameter und berechnet

$$\mu = \frac{1}{n} x^T z \ ,$$

auch bekannt als „duality measure“. Setzt man schließlich $\tau = \sigma \cdot \mu$, so ergibt sich der entsprechende Newton-Schritt:

$$\begin{pmatrix} A & 0 & 0 \\ 0 & A^T & I \\ Z & 0 & X \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta z \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ -XZe + \sigma\mu e \end{pmatrix}$$

$(\Delta x, \Delta y, \Delta z)^T$ ist somit ein Schritt in die Richtung $(x_{\sigma\mu}, y_{\sigma\mu}, z_{\sigma\mu}) \in C$ mit $x_{i,\sigma\mu} z_{i,\sigma\mu} = \sigma\mu$.

$$\begin{aligned} &\text{Falls } \sigma = 0 \quad \to \quad \text{Standard} - \text{Newton} - \text{Form auf KKT} \\ &\text{Falls } \sigma = 1 \quad \to \quad \text{Zentrierungsschritt} \end{aligned}$$

Betrachte nun x, y, z mit $\mu = \frac{1}{n} x^T z$ und $(x(t), y(t), z(t)) = (x, y, z) + t(\Delta x, \Delta y, \Delta z)$. Es stellt sich nun die Frage, wie sich die Wahl von σ auf unser $\mu(t)$ auswirkt:

$$n \cdot \mu(t) = x^T(t) z(t) = (x + t\Delta x)^T (z + t\Delta z) = x^T z + t(\Delta x^T + x\Delta z) + t^2 \Delta x^T \Delta z$$

Aus dem Newton-Schritt folgt nun

(i) $A\Delta x = 0$

(ii) $A^T \Delta y + \Delta z = 0$

$$\Rightarrow \underbrace{\Delta x^T A^T}_{=0} \Delta y + \Delta x^T \Delta z = 0 \Rightarrow \Delta x^T \Delta z = 0$$

Aus (i) folgt nun aus dem Newtonschritt: $Z\Delta x + X\Delta z = -XZe + \sigma\mu e \Rightarrow$

$$e^T Z \Delta x + e^T X \Delta z = -e^T XZe + \sigma\mu e^T e \Rightarrow z^T \Delta x + x^T \Delta z = -x^T z + \sigma\mu n$$

Somit ergibt sich nun für $\mu(t)$ das folgenden:

$$\mu(t) = \frac{x^T(t)z(t)}{n} = \frac{x^T z}{n} + \frac{t}{n}(-x^T\mu + n\sigma\mu) = \mu + t(-\mu + \sigma\mu) = (1 - t + \sigma t)\mu = (1 - t(1 - \sigma))\mu$$

$$\begin{array}{lll} \text{Falls } \sigma = 0 & \Rightarrow & \mu(t) = (1-t)\mu \quad \text{Reduzieren der Dualitätslücken} \\ \text{Falls } \sigma = 1 & \Rightarrow & \mu(t) = \mu \qquad\quad \text{weg vom Rand} \end{array}$$

Im IPM spricht man vom „trade-off“ zwischen der Zentrierung und der Reduzierung.

Rahmen-Algorithmus

Input: $x^0, y^0, z^0 \in \mathfrak{F}^0$ für $k = 0, 1, ...$ löse

$$\begin{pmatrix} A & 0 & 0 \\ 0 & A^T & I \\ Z^k & 0 & X^k \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta z \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ -X^k Z^k e + \sigma^k \mu^k e \end{pmatrix}$$

wobei $\sigma^k \in [0, 1],\ \mu = \frac{1}{n}(x^k)^T z^k$
Wähle nun t_k so, dass $(x^{k+1}, y^{k+1}, z^{k+1}) = (x^k, y^k, z^k) + t_k(\Delta x^k, \Delta y^k, \Delta z^k)$ mit $x^{k+1} > 0,\ z^{k+1} > 0$.
Die Wahl von t_k und σ_k ist an dieser Stelle äußerst wichtig!

Wie wichtig die Wahl von t_k und σ_k an dieser Stelle ist, zeigt allein die Tatsache, dass es in der Literatur viele verschiedene Ansätze darüber gibt. Die drei Hauptvarianten sollen nun kurz erläutert werden:

1. **Die Short-Step-Methoden**
   - σ_k wird nahe an 1 gewählt
   - Die Iterationen bleiben dicht am zentralen Pfad, μ wird sehr langsam reduziert, also kleiner.
   - $t_k \approx 1$
2. **Die Long-Step-Methoden**
   - σ_k wird nahe bei 0 gewählt
   - Lange Schritte zur Reduzierung von μ
   - Zentrales Problem ist hierbei die Wahl von t.
3. **Prädiktor-Korrektur-Methoden** Die Idee ist hierbei, dass man zwei Schritte pro Iteration durchführt. Hierbei ist die Prädiktur, der reine Newton-Algorithmus, die Korrektur ist der Zentrierungsschritt.

Die Theoretische Analyse, z.B. für das „Long-Step-Verfahren“: Hier ist die Nähe zum Rand von entscheidender Bedeutung.

$$N_{-\infty}(\gamma) = \{(x, y, z) : x_i z_i \geq \gamma\mu\} \quad , \quad \gamma \in]0, 1[$$

Algorithmus: Gegeben sind: $\gamma \in]0,1[,\ \sigma_{min},\ \sigma_{max}$ mit

$$0 < \sigma_{min} < \sigma_{max} < 1 \quad , \quad (x^0, y^0, z^0) \in N_{-\infty}(\gamma)$$

Für $k = 1, 2, ...$ wählt man $\sigma_k \in [\sigma_{min}, \sigma_{max}]$ und löst die lineare Gleichung für $(\Delta x^k, \Delta y^k, \Delta z^k)$. Nun wählt man $t_k \in]0,1[$ maximal, sodass $(x^k(t), y^k(t), z^k(t)) \in N_{-\infty}(\gamma)$.

Theorem 5.4 Gegeben sei ein $\epsilon > 0$ und ein $\gamma \in]0,1[$, sowie $(x^0, y^0, z^0) \in N_{-\infty}$. Außerdem sei $\mu \leq \frac{1}{e^k}$ für ein $k > 0$. Dann existiert ein Index K mit $K = \mathcal{O}(n\ log\ \frac{1}{n})$, sodass $\mu_s \leq \epsilon \quad \forall s \geq K$.

Da ein Beweis an dieser Stelle zu umfangreich wäre, wird hier darauf verzichtet. Als Abschluss des Kapitels sollen an dieser Stelle noch kurz die Unterschiede zwischen den beiden bisher kennengelernten Systemen angemerkt werden.

Innere-Punkt-Verfahren: Der Vorteil des IPM ist der durch die wenigen Iterationen bedingte polymiale Aufwand. Der Nachteil ist, dass diese wenigen Iterationen ziemlich teuer sind und außerdem kein „Warm-Start“, eine Verkürze Rechnung bei einer nur geringen Abänderung der Aufgabenstellung, möglich ist.

Simplex-Verfahren: Das Simplex-Verfahren ist genau das Gegenstück zum IPM. Hier sind die billigen Iterationen und die Möglichkeit der „Warm-Starts“ von Vorteil. Als nachteilig ist die große Anzahl der Iterationen zu sehen. Dadurch ergibt sich eine exponentielle Laufzeit. Das Simplex-Verfahren ist auch für die gemischt-ganzzahlige Optimierung geeignet.

6 Ganzzahlige Optimierung

Wie die bisherigen Verfahren der Lineare Optimierung beschäftigt sich die Ganzzahlige Optimierung mit der Optimierung linearer Zielfunktionen über einer Menge, die durch lineare Gleichungen und Ungleichungen eingeschränkt ist. Der Unterschied liegt darin, dass in der ganzzahligen Optimierung einige oder auch alle Variablen nicht beliebige, sondern nur ganzzahlige Werte annehmen dürfen. Hierdurch kommt dieses Verfahren den Problemen in der Betriebswirtschaft ziemlich nahe, was die folgenden Beispiele verdeutlichen sollen.

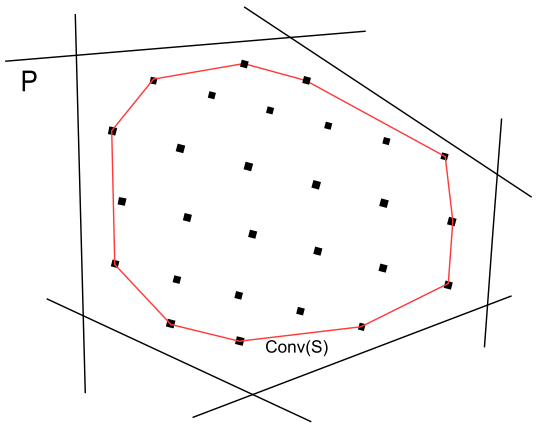
1. **Betriebsoptimierung:** „Kaufen oder Leihen“, Produktions-, Maschinenreihenfolge
2. **Planung:** Standort-, Portfolio- und Kapitaleinsatzplanung
3. **Design:** Kommunikations- und Transportnetzwerk, Produktionsanlagendesign oder das Design von Trennprozessen.

Ein Programm der „gemischt-ganzzahlige Optimierung“ (engl.: mixed-integer-programming, MIP) ist im Allgemeinen von der folgenden Gestalt:

$$(MIP) \qquad \begin{aligned} &\min_{x,y} \; c^T x + h^T y \\ &Ax + Gy \leq b \\ &x, y \geq 0 \;,\; x \in \mathbb{Z}^n, y \in \mathbb{R}^p \end{aligned}$$

Spezialfälle sind hierbei zum einen die ganzzahlige Optimierung (integer programming), bei der nur ganzzahlige Variablen auftauchen, oder zum anderen der Fall, wenn für x_i nur logische Entscheidungen zulässig sind, ergo $x_i \in \{0,1\} =: \mathbb{B}$. Somit ist $x \in \mathbb{B}^n$ ein binärer Vektor.

Es stellt sich natürlich nun die Frage, wie man in der ganzzahligen Optimierung den zulässigen Lösungsbereich beschreiben kann. In der linearen Optimierung hat man hierfür in der Regel einen Polyeder $P = \{x \in \mathbb{R}^n \mid Ax \leq b,\ x \geq 0\}$ gegeben. In der ganzzahligen Optimierung setzt man nun $S := P \cap \mathbb{Z}^n$ und betrachtet als zulässige Menge eines Programms schließlich $conv(S)$, die konvexe Hülle von S. Diese ist als konvexe Menge dann wieder ein Polyeder. Außerdem sind alle Eckpunkte, und somit auch alle potentiellen Lösungen, nur ganzzahlige Werte.



Beispiele

a) Das Problem bei begrenzter Platzkapazität (knapsack): Hier ist die Annahme, man hat n Projekte, wobei jedes Projekt j die Kosten a_j hat und den Nutzen c_j bringt. Insgesamt hat man ein Budget von b zur Verfügung. Es stelle sich nun die Frage, wie viele und vor allem welche

Projekte durchgeführt werden sollen. Deshalb berachtet man hierbei $x_i \in \mathbb{B} = \{0,1\}$. Mit diesen Voraussetzungen versucht man schließlich seinen Nutzen zu maximieren, also max $\sum_j c_j x_j$ unter der Voraussetzung $\sum_i a_i x_i \leq b$.

b) Überdeckungs-, Packungs- und Partionierungsprobleme: Sei $M = \{1, \ldots, m\}$. Weiter seien $M_1, ..., M_n$ Teilmengen von M. Eine Auswahl $F \subset \{1, \ldots, n\}$ der M_i nennet man nun

- „Überdeckung“ (cover) von M, falls $\bigcup_{i \in F} M_i = M$.
- „Packung“ (packing) von M, falls $\forall\ i, j \in I$ die Mengen M_i und M_j paarweise disjunkt.
- „Zerlegung“ (Partionierung), falls beides gilt.

Formulierung als ein boolsches (0-1) Problem: Man definiert $A \in \mathbb{B}^{m \times n}$ und $x \in \mathbb{B}^n$ mit

$$a_{ij} = \begin{cases} 1 & \text{, falls } i \in M_j \\ 0 & \text{, sonst} \end{cases} \quad , \quad x_j = \begin{cases} 1 & \text{, falls } j \in F \\ 0 & \text{, sonst} \end{cases} .$$

Die Auswahl F ist somit nun

- eine Überdeckung, wenn $Ax \geq \mathbb{1}$.
- eine Packung, wenn $Ax \leq \mathbb{1}$.
- eine Zerlegung, wenn $Ax = \mathbb{1}$.

Typische Anwendung: Standortplanung (facility location)
Annahme: $\{1, ..., n\}$ sind n mögliche Standorte für die Feuerwehr, $\{1, ..., m\}$ sind m Orte die zu erreichen sind und die Kosten den Standort j zu bauen betragen c_j. Von Standort j sind dann die Dörfer M_j in einer gewissen Zeit zu erreichen. Das Überdeckungsproblem lautet dann:

$$\min \sum c_j x_j \quad \text{unter} \quad Ax \geq \mathbb{1}$$

Ähnliche Probleme sind die Zuordnungen von Kunden zu Anlieferungsfahrten, Airline Crews zu Flügen oder die Zuordnung von Arbeitern zu den einzelnen Schichten.

6.1 Modellformulierung

Die Frage ist nun, was ist eine gute Modelformulierung? Anders als bei einem LP ist die Formulierung der Variablen und Ungleichungen entscheidend für eine gute Lösbarkeit. Das Problem hierbei ist, dass verschiedene Ungleichungen existieren, die die gleiche zulässige Menge des (IP) beschreiben. Im Allgemeinen ist ein (IP) nürlich auch von der Form

$$\text{(IP)} \qquad \min\ c^T x,\ Ax \leq b,\ x \geq 0,\ x \in \mathbb{Z}^n\ .$$

Für die wichtigen Algorithmen löst man jedoch zuerst das vereinfachte *„relaxierte Problem“*

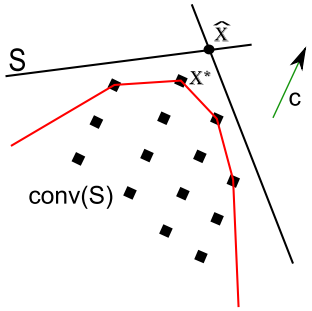
$$\text{(RIP)} \qquad \min\ c^T x,\ Ax \leq b,\ x \geq 0\ .$$

Theorem 6.1 Ist (RIP) unbeschränkt, so ist das (IP) entweder unzulässig oder unbeschränkt, also nicht lösbar.

Theorem 6.2 Für die Lösung $\hat{x}$ des (RIP) gilt im Vergleich zur Lösung x^* des (IP) immer

$$c^T \hat{x} \leq c^T x^* \ .$$

Das (RIP) liefert somit also eine untere Schranke für (IP).



Theorem 6.3 Jede zulässige ganzzahlige Lösung liefert eine obere Schranke für (IP).

Definition 6.4 *Die Formulierung (A, b) heißt* **stärker** *als die Formulierung (A', b'), falls gelten:*

(i) $P := \{x \mid Ax \leq b,\ x \geq 0\} \subset \{x \mid A'x \leq b',\ x \geq 0\} =: P'$

(ii) $\mathbb{Z}^n \cap P = \mathbb{Z}^n \cap P'$

Die Formulierung (A, b) heißt **exakt**, *wenn $P = conv(\mathbb{Z}^n \cap P)$.*

Beispiel $S = \{(0000), (1000), (0100), (0010), (0001), (0110), (0101), (0011)\} \subset \mathbb{B}^4$. Mögliche Formulierungen für S sind:

(a) $S_1 = \mathbb{B}^4 \cap \{x \mid 93x_1 + 49x_2 + 37x_3 + 29x_4 \leq 111\}$

(b) $S_2 = \mathbb{B}^4 \cap \{x \mid 2x_1 + x_2 + x_3 + x_4 \leq 2\}$

(c) $S_3 = \mathbb{B}^4 \cap \{x \mid 2x_1 + x_2 + x_3 + x_4 \leq 2,\ x_1 + x_2 \leq 1,\ x_1 + x_3 \leq 1,\ x_1 + x_4 \leq 1\}$

Hierbei sind $S_3 \subset S_1$ und $S_3 \subset S_2$. Somit ist S_3 die stärkste und demnach auch die Beste Formulierung für S.

Merke: Man darf nicht versuchen mit möglichst wenig Ungleichungen auszukommen. Stattdessen gibt es sogar eine Vielzahl von Algorithmen, die zusätzliche Ungleichungen konstruieren. Ein Beispiel hierfür ist der **Schnittebenenansatz**.

Ansatzmöglichkeiten: Will man ein ganzzahliges Programm lösen, so löst man immer zuerst das „relaxierte Problem“ (RIP). Hat man hierfür eine optimale Lösung $\hat{x}$ gefunden und ist diese nicht ganzzahlig, so gibt es zwei Varianten:

1. Hinzufügen engerer Ungleichungen
2. Branch-and-Bound: Fallunterscheidungen (löse das Problem links, rechts, unten, oben)

6.2 Schnittebenenverfahren (Cutting Planes)

Das Prinzip dieser Verfahren besteht darin, anstelle des ganzzahligen Problems eine Folge linearer Probleme zu lösen. Ist die Lösung des jeweiligen linearen Problems nicht ganzzahlig, so wird eine Schnittebene berechnet und diese als zusätzliche Restriktion zu den Nebenbedingungen hinzugefügt. Dann wird erneut das lineare Programm gelöst. Dies wird so lange wiederholt, bis ein lineares Programm eine ganzzahlige optimale Lösung hat oder bis feststeht, dass keine ganzzahlige Lösung existiert. Dieses Verfahren endet in endlich vielen Schritten, falls der zulässige Lösungsraum beschränkt ist und ein Kreisen des Verfahrens verhindert wird.

Bemerkung Es existiert ein Beispiel („matching problem“), sodass P_I ($:= conv(S)$) nur mit $\mathcal{O}(2^n)$ Ungleichungen beschrieben werden kann, das heißt, eine exakte Formel ist nur im Spezialfall möglich.

Beispiel Sei $A \in \mathbb{Q}^{m\times n}, b \in \mathbb{Q}^m$ (z.B. 64 Bit Zahlen). Finde eine positive Linearkombination ($\lambda \in \mathbb{Q}^n, \lambda \geq 0$) mit $\underbrace{\lambda^T A}_{\pi^T} x \leq \underbrace{\lambda^T b}_{\pi_0}$ $\forall x \in S$. Somit ist für $\pi \in \mathbb{Z}^n$: $\pi^T x \leq \pi_0$. Dann ist

$$(*) \quad \pi^T x \leq \lfloor \pi_0 \rfloor$$

eine stärkere Ungleichung für S. Das Problem ist nur die Konstrukion von λ. Für alle Räume mit $n > 2$ werden weitere Bedingungen notwendig. $\lfloor - \rfloor$ ist hierbei die Abrundung.

Definition 6.5

1. $e^0 := \{$*alle Ungleichungen in* $Ax \leq b\}$, $P^0 = \{x | Ax \leq b\}$
2. $e^1 := \{$*alle Ungleichungen, die auf* $(*)$ *aus* e^0 *erzeugbar sind* $\}$, $P^1 = \{x | x$ erfüllt $e^1\}$
3. *Definiere weiter rekursiv* e^{k+1} *aus* e^k *und* $P^{k+1} = \{x | x$ erfüllt $e^{k+1}\} \subset P^k$
4. $Cl(e^0) := \bigcup_{i=1}^{\infty} e^i$ *, wobei* $Cl(e^0)$ *- Closure of* e^0

Mit diesen Definitionen gilt dann der folgende Satz:

Satz 6.6 Seien A, b wie in dem obigen Beispiel und $P = \{x \mid Ax \leq b, x \geq 0\}$ beschränkt. Dann gelten folgende Aussagen:

(i) alle ganzzahligen Ungleichungen $d^T x \leq d_0$ mit $d \in \mathbb{Z}^n$, $d_0 \in \mathbb{Z}$, die $P_I = conv(\mathbb{Z}^n \cap P)$ charakterisieren, sind in Cl.

(ii) Es existiert ein $k < \infty$ mit $P^k = P_I$ und endlich viele Ungleichungen in P^k charakterisieren P_I.

6.2.1 Gomory-Cuts

Im Jahre 1958 während seines Aufenthaltes in Princeton entwickelte Ralph Gomory das erste allgemein einsetzbare Schnittebenenverfahren. Die Idee ist, dass bei jeder Iteration eine LP-Relaxierung gelöst wird. Falls das Optimum $\hat{x}$ nicht ganzzahlig ist, wird eine Ungleichung $\pi^T x \leq \pi_0$ bestimmt, die das Optimum $\hat{x}$ abschneidet, jedoch von allen $x \in Conv(P \cap \mathbb{Z}^n)$ erfüllt wird (Seperation).

Betrachtet man nun den Spezialfall $\{\min c^T x,\ Ax = b,\ x \geq 0,\ x \in \mathbb{Z}^n\}$, wobei c, A, b ganzzahlige Daten sind, so gilt der folgende Satz:

Satz 6.7 Sei B eine optimale Basis des Programms $\{\min\ c^T x,\ Ax = b,\ x \geq 0\}$. Seien weiter $\bar{A} := A_B^{-1} A_N,\ \bar{b} := A_B^{-1} b$, so gelten für alle $x \in P \cap \mathbb{Z}^n$ die folgenden Ungleichungen:

$$\sum_{j \in N} (\lfloor \bar{a}_{ij} \rfloor - \bar{a}_{ij}) x_j \leq \lfloor \bar{b}_i \rfloor - \bar{b}_i \qquad \forall\ i \in B$$

Beweis Es ist $x_B = A_B^{-1}(b - A_N x_N) = \bar{b} - \bar{A} x_N$ oder

$$(i) \quad x_i + \sum_{j \in N} \bar{a}_{ij} x_j = \bar{b}_i \quad , \quad i \in B\ .$$

Da $x_N \geq 0$ folgt $x_i + \sum_{j \in N} \lfloor \bar{a}_{ij} \rfloor x_j \leq \bar{b}_i$. Mit $x \in P \cap \mathbb{Z}^n$ ist hierbei die linke Seite ganzzahlige, wodurch die Verschärfung

$$(ii) \quad x_i + \sum_{j \in N} \lfloor \bar{a}_{ij} \rfloor x_j \leq \lfloor \bar{b}_i \rfloor \quad , \quad i \in B$$

möglich ist. Subtrahiert man nun (i) von (ii), so folgt die Behauptung. ■

Nun betrachtet man die gefundene optimale Lösung $\hat{x}$ des relaxierten Problems. Ist diese nicht ganzzahlige, so existiert ein $i_0 \in B$ mit $\hat{x}_{i_0} = \bar{b}_{i_0} \notin \mathbb{Z}$. Somit folgt

$$\lfloor \bar{b}_{i_0} \rfloor - \bar{b}_{i_0} < 0 = \sum_{j \in N} (\lfloor \bar{a}_{i_0,j} \rfloor - \bar{a}_{i_0,j}) \underbrace{x_j}_{=0}\ .$$

Die Lösung $\hat{x}$ erfüllt somit nicht die Ungleichung aus Satz 6.7, woraus direkt folgt:

Korrolar 6.8 Seien für ein $i \in B$ $f_{ij} := \bar{a}_{ij} - \lfloor \bar{a}_{ij} \rfloor,\ j \in N$ und $f_{i0} := \bar{b}_i - \lfloor \bar{b}_i \rfloor,\ f_{i0} \neq 0$. Dann ist

$$\sum_{j \in N} (-f_{ij}) x_j \leq -f_{i0}$$

eine Ungleichung, die für alle $x \in Conv(P \cap \mathbb{Z}^n)$ zulässig ist, aber von $\hat{x}$ nicht erfüllt wird.

Der Gomory-Algorithmus

Der Gomory-Algorithmus verwendet das primale und duale Simplexverfahren.

1. Zuerst bestimmt man eine zulässige relaxierte Lösung x mit Hilfe der Phase I des primalen Simplex-Verfahrens. Existiert keine zulässige Lösung, ist also das (RIP) unzulässig, so ist auch das (IP) unzulässig. Hat man jedoch eine zulässige Lösung gefunden, so bestimmt man nun mit Hilfe der Phase II des primalen Simplex-Verfahrens eine optimale relaxierte Basislösung $\hat{x}$. Existiert eine soche nicht, so ist auch das (IP) nicht lösbar.

2. Mit der Basis des (RIP) prüft man nun, ob

$$\hat{x}_B = A_B^{-1} b = \bar{b} \in \mathbb{Z}^m \ .$$

   Ist dies der Fall, so hat man mit $x^* := \hat{x}$ schon die optimale, ganzzahlige Lösung des (IP) gefunden. Ist dies nicht der Fall, so setzt man den Algorithmus fort.

3. Da $\hat{x} \notin \mathbb{Z}^n$, existiert mindestens ein $i \in B$ mit $\hat{x}_i = \bar{b}_i \notin \mathbb{Z}$. Für dieses (oder evtl. auch mehrere) i leitet man nun einen Gomory-Schritt her:

$$\sum_{j \in N} (-f_{ij}) x_j \leq -f_{i0}$$

4. Füge die Schnitte dem LP hinzu. Hierbei wird für jeden Schnitt eine neue Schlupfvariable eingeführt, die in $\hat{x}$ unzulässig ist.

5. Nun wendet man bis zur primalen Zulässigkeit das duale Simplex-Verfahren an. Ist das duale LP unbeschränkt, so ist das (IP) unzulässig. Ist dies nicht der Fall, so führt man den Algorithmus bei Schritt 2 fort.

Beispiel Gegeben sei das folgende lineare Problem: max $x_1 + 2x_2$ mit den Restriktionen

$$\begin{array}{rcr} x_1 \leq 4 & & x_1 + x_3 = 4 \\ 2x_1 + x_2 \leq 10 & \Leftrightarrow & 2x_1 + x_2 + x_4 = 5 \\ -x_1 + x_2 \leq 5 & & -x_1 + x_2 + x_5 = 5 \\ x_1, x_2 \geq 0 & & x_1, x_2, x_3, x_4, x_5 \geq 0 \end{array}$$

Durch den Simplex-Algorithmus erhält man die Lösung $\hat{x} = (\frac{5}{3}, \frac{20}{3}, \frac{7}{3}, 0, 0)^T$ mit der Basis $B = \{1, 2, 3\}$. Diese ist jedoch nicht ganzzahlig. Für eine oder auch mehrere nichtganzzahlige Komponenten der Lösung $\hat{x}$ kann man nun einen Gomery-Cut durchführen. Hierfür betrachtet man zunächst

$$\bar{A} = A_B^{-1} A_N = \begin{pmatrix} \frac{1}{3} & -\frac{1}{3} \\ \frac{1}{3} & \frac{2}{3} \\ -\frac{1}{3} & \frac{1}{3} \end{pmatrix} \quad , \quad \bar{b} = A_b^{-1} b = \begin{pmatrix} \frac{5}{3} \\ \frac{20}{3} \\ \frac{7}{3} \end{pmatrix} \ .$$

Die Spalten von $\bar{A}$ beziehen sich auf x_4, x_5 und die Zeilen von $\bar{b}$ auf die Variablen in der Basis, also x_1, x_2, x_3. In diesem Beispiel beschränken wir uns zunächst auf die Komponente x_1.

Gesucht wird somit nun der Gomery-Schnitt für $i = 1$:

$$-(\bar{a}_{14} - \lfloor \bar{a}_{14} \rfloor)x_4 - (\bar{a}_{15} - \lfloor \bar{a}_{15} \rfloor)x_5 \leq -(\bar{b}_1 - \lfloor \bar{b}_1 \rfloor)$$

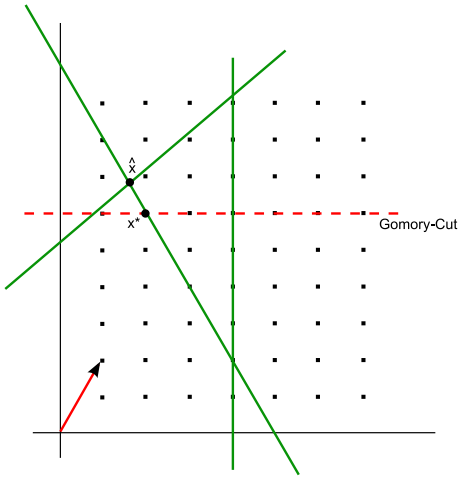
$$\Leftrightarrow \quad -\frac{1}{3}x_4 - \frac{2}{3}x_5 \leq -\frac{2}{3}$$

In diese Gleichung setzt man nun die Restriktionen für x_4 und x_5 ein. Es folgt:

$$-\frac{1}{3}(10 - 2x_1 - x_2) - \frac{2}{3}(5 + x_1 - x_2) \leq -\frac{2}{3}$$

$$\Leftrightarrow \quad -10 + 2x_1 + x_2 - 10 - 2x_1 + 2x_2 \leq -2$$

$$\Leftrightarrow \quad 3x_2 \leq 18 \quad \Leftrightarrow \quad x_2 \leq 6$$



Hiermit hat man schließlich eine zusätzliche Restriktion, welche die optimale Lösung $\hat{x}$ des (RIP) abschneidet, gefunden. Fügt man diese den Restriktionen des (RIP) an, so bleibt $\hat{x}$ jedoch weiterhin dual zulässig. Deshalb kann man nun mit dem dualen Simplex-Verfahren nachoptimieren. In unserem Beispiel erhält man hierdurch schon die optimale ganzzahlige Lösung $x^* = (2, 6)^T$ des (IP). Auch an der graphischen Lösung wird hierbei sehr anschaulich verdeutlicht, wie der Gomery-Cut zum Finden der optimalen Lösung des (IP) beigetragen hat.

Bemerkung Unter der Voraussetzung, dass A und b nur ganzzahlige Werte enthalten, kann man beiweisen, dass der Gomory-Algorithmus bei Beachtung der bekannten „Anti-Schleifen-Regeln“ immer terminiert.

6.3 Branch and Bound

Wie der Name schon sagt, setzt sich das Branch-&-Bound-Verfahren aus zwei Schritten zusammen:

Branch (Verzweigung) Der Branch-Schritt dient dazu, das vorliegende Problem in zwei oder mehr Teilprobleme aufzuteilen, die eine Vereinfachung des ursprünglichen Problems darstellen. Durch rekursives Ausführen des Branching-Schritts für die erhaltenen Teilprobleme entsteht eine Baum-Struktur. Es gibt verschiedene Auswahlverfahren für die Wahl des nächsten zu bearbeitenden Knotens im Branching-Baum, die unterschiedliche Zielsetzungen haben. Im Folgenden werden drei häufig verwendete Verfahren beschrieben:

- Tiefensuche: Von den noch nicht bearbeiteten Teilproblemen wird das gewählt, welches als letztes eingefügt wurde (*Last In - First Out*). Mit dieser Auswahlregel arbeitet sich das Verfahren im Baum möglichst schnell in die Tiefe, so dass sehr schnell eine zulässige Lösung erlangt wird, über deren Qualität jedoch nichts ausgesagt werden kann.

- Breitensuche: Von den noch nicht bearbeiteten Teilproblemen wird das gewählt, welches als erstes in den Baum eingefügt wurde (*First In - First Out*). Bei Verwendung dieser Auswahlregel werden die Knoten im Baum pro Ebene abgearbeitet, bevor tiefer gelegene Knoten betrachtet

werden. Eine zulässige Lösung wird in der Regel erst relativ spät erhalten, hat aber tendenziell einen guten Lösungswert.

- Bestensuche: Von den noch nicht bearbeiteten Teilproblemen wird das gewählt, welches die beste untere Schranke vorweist. Durch diese qualitativ arbeitende Auswahlregel wird versucht, direkt einen sehr guten Lösungswert als erste Lösung zu erhalten.

Bound (Abgrenzung) Der Bound-Schritt hat die Aufgabe, bestimmte Zweige des Baumes „abzuschneiden“, d.h. in der weiteren Berechnung nicht mehr zu betrachten, um so den Rechenaufwand zu begrenzen. Dies erreicht der Algorithmus durch Berechnung und Vergleich zweier Schranken. Der Weg von der Wurzel des Baumes zu einem Teilproblem bildet die untere Schranke (*„lower bound“*) dieses Teilproblems - auf welche Weise sich auch der weitere Weg durch den Baum gestaltet, der Weg zum Teilproblem ist festgelegt. Als obere Schranke dient eine vorerst optimale zulässige Lösung. Diese erhält man durch Berechnung eines kompletten Zweiges (von der Wurzel zu einem Blatt des Entscheidungsbaumes) - ebenso ist es aber auch möglich, eine Heuristik vor dem eigentlichen Branch-and-Bound Verfahren zu berechnen und die erhaltene Lösung als obere Schranke zu verwenden.
Übersteigt nun die untere Schranke in einem Teilproblem die obere Schranke, so kann der aktuell betrachtete Teilbaum „abgeschnitten“ werden, da es mindestens eine bessere zulässige Lösung gibt, die nicht mehr erreicht werden kann. Sollte hingegen eine Komplettlösung berechnet werden, die besser als die obere Schranke ist, so wird diese ersetzt und als neue optimale Lösung vorgehalten.

Branch-&-Bound-Algorithmus

Als Input hat man bei der Branch-&-Bound-Methode ein gemischt ganzzahliges Problem

$$(MIP) \qquad \begin{array}{c} \min\ c^T x \\ Ax \leq b \\ x \geq 0,\ x_i \in \mathbb{Z},\ i \in N \end{array}$$

gegeben. Hierbei sind wie gewohnt $A \in \mathbb{R}^{m \times n},\ x \in \mathbb{R}^n,\ b \in \mathbb{R}^m$. Als Output erhält man entweder eine Lösung des (MIP), oder auch den Nachweis der Unlösbarkeit. Das Verfahren ansich setzt sich nun aus den folgenden drei Schritten zusammen:

(0) Zu Beginn des Verfahrens setzt man $P_0 := \{x \in \mathbb{R}^n \mid Ax \leq b,\ x \geq 0\}$ und $k = 0$. Weiter ist $K = \{P_0\}$ die Liste der noch zu bearbeitenden Knoten. Hierbei ist k der Index des Knotens, der zuletzt der Menge K hinzugefügt wurde.
Weiter sei $\bar{x}$ (zu Beginn undefiniert) die beste Zulässige Lösung und $\bar{L} := c^T \bar{x}$ die beste obere Schranke. Zu Beginn des Verfahrens gilt, da $\bar{x}$ nicht bekannt, $\bar{L} = \infty$.

(1) Ist $K = \varnothing$, so bricht man an dieser Stelle das Verfahren ab. Ist hierbei $\bar{L} = \infty$, so hat das (MIP) keine Lösung. Ansonsten ist $\bar{x}$ die optimale Lösung des (MIP) und es gilt

$$c^T \bar{x} = \bar{L}\ .$$

(2) Da $K \neq \varnothing$, kann man an dieser Stelle einen Branch-Schritt, also eine Verzweigung, durchführen.

Man wählt einen Knoten $P_j \in K$ und löst das relaxierte Problem

$$(LP_j) \qquad \begin{array}{c} \min\ c^T x \\ x \in P_j \end{array} \quad .$$

(a) Ist das relaxierte Problem (LP_j) unlösbar, so braucht man diesen „Zweig“ nicht weiter betrachten und kann zu Schritt (3) wechseln.

(b) Hat (LP_j) eine Lösung x^*, so setzt man $L^* = c^T x^*$. In diesem Fall ist eine weitere Unterscheidung notwendig:

(i) Gilt an dieser Stelle $L^* \geq \bar{L}$, so braucht man auch in diesem Fall den Knoten P_j nicht weiter zu betrachten, da hier keine bessere Lösung als die bisher beste Lösung $\bar{x}$ gefunden werden kann.

(ii) Gilt jedoch $L^* < \bar{L}$, so hat man mit x^* evtl. eine neue beste zul. Lösung gefunden. Dies ist genau dann der Fall, wenn

$$x_i^* \in \mathbb{Z} \quad \forall\, i \in N \ ,$$

da in diesem Fall x^* zulässig in (MIP) ist. Man setzt $\bar{x} := x^*$ und $\bar{L} := L^*$ und setzt das Verfahren mit Schritt (3) fort.

Ist jedoch x^* in (MIP) nicht zulässig, d.h. $\exists\ i \in N$ mit $x_i^* \notin \mathbb{Z}$, so führt man eine weitere Verzweigung durch. Hierfür fügt man die zwei Knoten

$$P_{k+1} := P_j \cap \{x_i \leq \lfloor x_i^* \rfloor\} \quad , \quad P_{k+2} := P_j \cap \{x_i \geq \lceil x_i^* \rceil\}$$

der Liste K hinzu und setzt $k = k+2$. Nun kann das Verfahren bei Schritt (3) fortgesetzt werden.

(3) Entferne den Knoten P_j aus der Liste K und gehe wieder zu Schritt (1).

Beispiel Es sei gegeben das ganzzahlige Optimierungsproblem

$$\begin{array}{c} \min\ 7x_1 + 3x_2 + 4x_3 \\ x_1 + 2x_2 + 3x_3 - x_4 = 8 \\ 3x_1 + x_2 + x_3 - x_5 = 5 \\ x \geq 0\ ,\ x \in \mathbb{Z}^5 \end{array} \quad .$$

Löst man hierzu das revidierte Programm, also (LP_0), so erhält man

$$x_0^* = \left(\frac{2}{5}, \frac{19}{5}, 0, 0, 0\right)^T \quad , \quad L_0^* = 14,2 \ .$$

Der Baum erhält somit die Schranke $\bar{L} \geq 15$, da hierbei nur der nächst größere ganzzahlige Wert betrachtet werden muss. x_0^* ist nun jedoch nicht ganzzahlig. Deshalb führt man an dieser Stelle eine weitere Verzweigung durch. Es ergibt sich

$$P_1 = P_0 \cap \{x_2 \leq 3\} \quad , \quad P_2 = P_0 \cap \{x_2 \geq 4\} \ .$$

Als Liste der noch zu bearbeitenden Knoten erhält man demnach nun $K = \{P_1, P_2\}$. Löst man hierzu nun (LP_1), so erhält man $x_1^* = (\frac{1}{2}, 2, \frac{1}{2}, 0, 0)^T$ und $L_1^* = 14,5$. Da man bis jetzt nun lediglich die Information $\bar{L} \geq 15$ hat, gilt in diesem Moment evtl. $L_1^* < \bar{L}$. Deshalb muss auch dieser Knoten weiter untersucht werden. Die Verzweigung

$$P_3 = P_1 \cap \{x_1 \leq 0\} \quad , \quad P_4 = P_1 \cap \{x_1 \geq 1\}$$

ergibt somit nun $K = \{P_2, P_3, P_4\}$. Hierbei liefert (LP_3) nun als optimale Lösung $x_3^* = (0, 3, 2, 4, 0)^T$ mit $L_3^* = 17$. Da nun $x_3^* \in \mathbb{Z}^5$, hat man hiermit eine neue beste Lösung gefunden. Man setzt $\bar{x} := x_3^*$ und $\bar{L} = L_3^* = 17$ und erhält $K = \{P_2, P_4\}$. Löst man nun (LP_4), um einen weiteren Knoten der Liste K abzuarbeiten, erhält man $x_4^* = (1, 3, \frac{1}{3}, 0, \frac{4}{5})^T$ mit $L_4^* = 17\frac{1}{3}$. Da dieser Wert nun echt schlechter als der bisherige Optimalwert $\bar{L}$ ist, braucht man auch diesen Knoten nicht weiter zu betrachtet, da sich hier keine bessere ganzzahlige Lösung finden lässt. Schließlich betrachtet man den Knoten P_2. Löst man (LP_2), so erhält man $x_2^* = (\frac{1}{3}, 4, 0, \frac{1}{3}, 0)^T$ und $L_2^* = 14\frac{1}{3}$. Es gilt also $L_2^* < \bar{L}$. Der Knoten P_2 muss demnach weiter untersucht werden. Da nun $x_2^* \notin \mathbb{Z}^5$, führt man eine weitere Zerlegung

$$P_5 = P_2 \cap \{x_1 \leq 0\} \quad , \quad P_6 = P_2 \cap \{x_1 \geq 1\}$$

durch. (LP_5) liefert hierbei $x_5^* = (0, 5, 0, 2, 0)^T$ mit $L_5^* = 15 < \bar{L}$. Im Knoten P_5 hat man somit eine neue beste ganzzahlige Lösung gefunden. Deshalb setzt man $\bar{x} := x_5^*$ und $\bar{L} := L_5^* = 15$. Außerdem ist es an dieser Stelle schon nicht mehr nötig den Knoten P_6 zu betrachten, da auf jeden Fall gelten muss $L_6^* \geq L_2^* = 14\frac{1}{3}$. Somit hat jede ganzzahlige Lösung, die man im Zweig unter P_6 finden kann eine untere Schranke von höchstens 15, also nicht besser als $\bar{x}$. Da somit kein weiterer Knoten zum Überprüfen in der Liste vorhanden ist, hat man das (IP) gelöst und erhält

$$x^{opt} = \begin{pmatrix} 0 \\ 5 \\ 0 \\ 2 \\ 0 \end{pmatrix} \quad \text{mit} \quad c^T x^{opt} = 15 \ .$$

Index