Lineare Optimierung

Vorlesung von

Prof. Dr. Stefan Körkel
Fakultät Informatik und Mathematik
OTH Regensburg

Literaturempfehlungen:

- S. Körkel (Mann-)Skript Lineare Optimierung, SS 2011, Universität Heidelberg
- R. Hornung, Lineare Optimierung, Logos Verlag
- M. Padberg, Linear Optimization and Extensions, Springer
- H. A. Eiselt, C.-L. Sandblom, Linear Programming and its Applications, Springer
- R. J. Vanderbei, Linear Programming, Springer
- D. G. Luenberger, Y. Ye, Linear and Nonlinear Programming, Springer
- M. Grötschel, Skript Lineare Optimierung, TU Berlin
- E. Kostina, Skript Lineare Optimierung, Uni. Marburg
- Kapitel über Lineare Optimierung in Lehrbüchern über Operations Research, Optimierung oder Numerik

Überblick über die Vorlesung

1. Operations Research und Optimierung
2. Lineare Optimierungsprobleme
3. Modellierung, Beispielprobleme
4. Graphische Lösung
5. Polyeder und ihre Eigenschaften
6. Dualität und Optimalitätssätze
7. Die Simplex-Methode
8. Software
9. Duales Simplex und Re-Optimierung
10. Innere-Punkt-Methoden

Praktische Aspekte, Übungen

- Modellierung
- Lösung mit Software
- Algorithmen

1. Operations Research und Optimierung

Operations Research: (Wikipedia)

Entwicklung und Einsatz quantitativer Modelle und Methoden zur Entscheidungsunterstützung

Schritte:

1. Erkennen und Analysieren eines Problems
2. Bestimmen von Zielen und Handlungsmöglichkeiten
3. Beschreibung als mathematische Modell
4. Beschaffung von Daten
5. Lösungsfindung durch mathematische Methoden
6. Bewertung der Lösung

Wichtige Begriffe: – Gegenstände dieser Vorlesung

- Optimierungsprobleme
- mathematische Modelle
- mathematische Methoden und Algorithmen
- numerische Lösung
- Analyse der Lösung

Mathematische Optimierungsprobleme

Bestandteile:

- Ich will etwas minimieren oder maximieren, und zwar eine Größe: Zielfunktion
- Ich kann dazu Werte von Variablen wählen.
- Es müssen Nebenbedingungen erfüllt sein.

Allgemein:

$$\max_{x \in \Omega} f(x) \qquad f(x) \in \mathbb{R} \qquad (1.1)$$

Bemerkung 1.1:

Wir können o.B.d.A. jedes Optimierungsproblem als Maximierungsproblem schreiben, bei Minimierungsproblemen $\min f(x)$ können wir nämlich genauso gut $-f(x)$ maximieren.

Varianten von Optimierungsproblemen

- Variablen aus unendlich-dimensionalem Raum
  → Optimierung in Funktionenräumen
- Variablen aus endlich-dimensionalem Raum

- unendlich-viele Nebenbedingungen
  → (semi-) infinite Optimierung
- endlich-viele Nebenbedingungen

- Problem enthält stochastische Größen
  → stochastische Optimierung
- Problem enthält nur deterministische Größen

- Variablen aus diskretem Raum ($\mathbb{Z}^n$)
  → diskrete / kombinatorische Optimierung
- Variablen aus kontinuierlichem Raum ($\mathbb{R}^n$)
  → kontinuierliche Optimierung

- Zielfunktion und Nebenbedingungen sind nicht linear → nichtlineare Optimierung

- Zielfunktion und Nebenbedingungen sind linear
  → Lineare Optimierung

- Probleme haben keine Nebenbedingungen
  → unbeschränkte Optimierung
- Probleme haben Nebenbedingungen
  → beschränkte Optimierung

In dieser Vorlesung behandeln wir die unterstrichenen Varianten, unsere Probleme haben also die Gestalt:

$$
\begin{aligned}
\max \quad & c^T x = \sum_{j=1}^{n} c_j x_j && x \in \mathbb{R}^n \\
\text{s.t.} \quad & a_1^T x = \sum_{j=1}^{n} a_{1j} x_j = b_1 && \text{oder} \leq b_1 \text{ oder} \geq b_1 \\
& \vdots && \vdots \\
& a_m^T x = \sum_{j=1}^{n} a_{mj} x_j = b_m && \text{oder} \leq b_m \text{ oder} \geq b_m
\end{aligned}
\tag{1.2}
$$

Die Nebenbedingungen können Gleichungs- oder Ungleichungsnebenbedingungen sein.

Zulässigkeit und Optimalität

Definition 1.2: (Zulässigkeit)

Wir betrachten das Optimierungsproblem (1.1).
Ein Punkt x heißt zulässig für (1.1), wenn er die Nebenbedingungen erfüllt:

$x \in \Omega$.

Wir Ω die Menge aller zulässigen Punkte.

Beispiel:

Lineares Optimierungsproblem (1.2).

$$\Omega = \{ x \in \mathbb{R}^n : a_i^T x = b_i \text{ oder } \leq b_i \text{ oder } \geq b_i, \; i = 1, \dots, m \}$$

Definition 1.2: (Optimalität)

Ein Punkt x^* heißt maximal für Problem (1.1), wenn er

1. zulässig ist: $x^* \in \Omega$
2. $f(x^*) \geq f(x)$ für alle $x \in \Omega$ (1.3)
   (globales Maximum)

bzw. $f(x^*) \geq f(x)$ für alle $x \in \Omega$, x in einer Umgebung von x^* (2.4)

(lokales Maximum)

Gilt jeweils $f(x^*) > f(x)$ für $x \neq x^*$, heißt das Maximum strikt.

Wir werden sehen: Bei linearen Optimierungsproblemen ist jedes lokale Maximum auch ein globales Maximum.

Mathematische Optimierung

In der mathematischen Optimierung werden Verfahren betrachtet und entwickelt, die Lösungen von (1.1) berechnen, die beweisbar optimal sind, entweder durch direkten Nachweis von (2.3) oder (2.4) oder durch Nachweis von hinreichenden und notwendigen Optimalitätsbedingungen.

Im allgemeinen Sprachgebrauch wird der Begriff „optimieren" meist nicht so streng gebraucht, er meint oft nur „verbessern".

In der Mathematik nennt man Verfahren zur „Lösung" von Optimierungsproblemen ohne Beweisbarkeit der Optimalität Heuristiken.

Prominentes Beispiel: Monte-Carlo-Verfahren

2. Lineare Optimierungsprobleme

Optimierung (d.h. Maximierung oder Minimierung)

- eine lineare Zielfunktion
- unter endlich vielen linearen Nebenbedingungen.

Bezeichnung:

LOP: Lineare Optimierung

LP: Lineare Programmierung (Programmierung = Planung)

Geschichte der linearen Optimierung

1939 L.W. Kantorowitsch: „Mathematische Methoden für die Organisation und Planung der Produktion" (Sowjetunion) Nobelpreis WW 1975

1941 F.L. Hitchcock: Transportproblem (USA)

1947 G. Dantzig: Simplex-Verfahren
zunächst für Optimierung militärischer Einsätze

ab 1950er Anwendung der LP in der Wirtschaft,

insbesondere Ölraffinerien

1979 L. Khachiyan: Ellipsoidmethode: $LP \in P$

1984 N. Karmarkar: Innere-Punkte-Verfahren

Sowohl Simplex-Algorithmus als auch IP-Verfahren sind nach wie vor Gegenstand der aktuellen Forschung.

Heute: In praktischen Anwendungen sind die LP-Matrizen dünn besetzt, es können Lineare Programme mit Hunderttausenden Variablen innerhalb von Minuten bis Stunden optimal gelöst werden.

Notationen

a) $x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \in \mathbb{R}^n$ Spaltenvektor

$x^T = (x_1, \dots, x_n)$ Zeilenvektor

$y^T x = \sum_{i=1}^{n} y_i x_i$ Skalarprodukt

$\|x\| = \sqrt{x^T x}$ Norm

$e_j = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \leftarrow j$ Einheitsvektor

$0 = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}$ Nullvektor $\quad \mathbb{1} = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}$ Einsvektor

b) $A = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix} = (a_{ij})_{\substack{i=1\dots m \\ j=1\dots n}}$ Matrix

$A_{\cdot j} = \begin{pmatrix} a_{1j} \\ \vdots \\ a_{mj} \end{pmatrix}$ j-te Spalte

$A_{i\cdot} = (a_{i1}, \dots, a_{in})$ i-te Zeile

Zeilenindexvektor $I = (i_1, \dots, i_p)$

Spaltenindexvektor $J = (j_1, \dots, j_q)$

$A_{IJ} = (a_{ij})_{\substack{i \in I \\ j \in J}}$ Teilmatrix

$0 = 0_{m \times n}$: Nullmatrix $\quad E = E_{m \times n}$: Einsmatrix

$I = \begin{pmatrix} 1 & & \\ & \ddots & \\ & & 1 \end{pmatrix} = I_{m \times m}$ Einheitsmatrix

c) $x_1, \dots, x_k \in \mathbb{R}^n \qquad \lambda_1, \dots, \lambda_k \in \mathbb{R}$

Linearkombination: $\sum_{i=1}^{k} \lambda_i x_i$

Konische Kombination: $\sum_{i=1}^{k} \lambda_i x_i \qquad \lambda_i \geq 0$

Affine Kombination: $\sum_{i=1}^{k} \lambda_i x_i, \ \sum_{i=1}^{k} \lambda_i = 1$

$= x_0 + \sum_{i=1}^{k} \mu_i (x_i - x_0)$

Konvexe Kombination: $\sum_{i=1}^{k} \lambda_i x_i$, $\lambda_i \geq 0$, $\sum_{i=1}^{k} \lambda_i = 1$

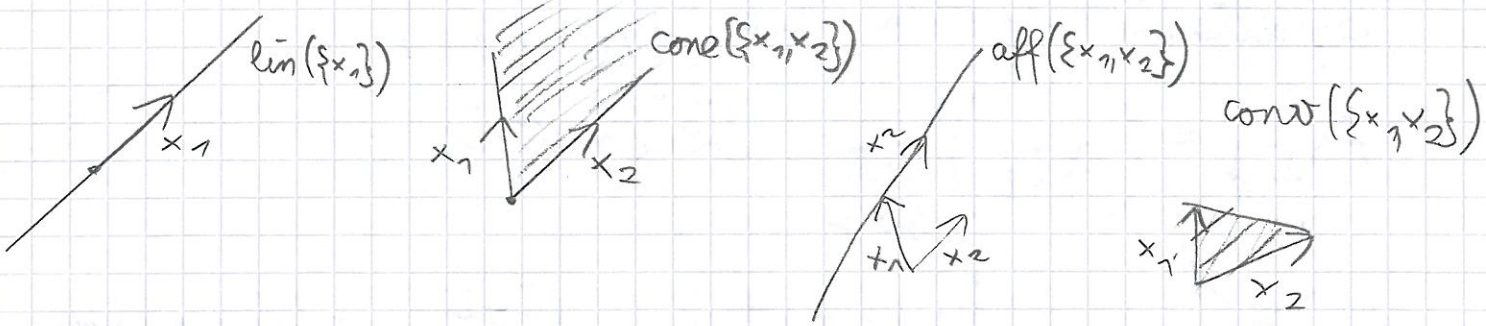
$S = \{x_1, \dots, x_k\}$

$\text{lin}(S)$: lineare Hülle: alle Linearkombinationen von Vektoren in S

$\text{cone}(S)$: konische Hülle: alle konischen Kombinationen

$\text{aff}(S)$: affine Hülle: alle affinen Kombinationen

$\text{conv}(S)$: konvexe Hülle: alle Konvexkombinationen



lineare Raum Kegel affine Raum konvexer Kegel

d) $x, y \in \mathbb{R}^n$

$x \geq y \iff x_i \geq y_i \quad \forall i = 1, \dots, n$

Beispiel: Ölraffinierung

- Rohöl R → schweres Öl S
  → mittelschweres Öl M
  → leichtes Öl L
- Verfahren 1: 10 ME R → 2 ME S, 2 ME M, 1 ME L
  (ME: Mengeneinheiten) Kosten: 3 GE
  (GE: Geldeinheiten)
  Verfahren 2: 10 ME R → 1 ME S, 2 ME M, 4 ME L
  Kosten: 5 GE
- Lieferverpflichtungen: 3 ME S, 5 ME M, 4 ME L
- Variablen: Wieviele Mengeneinheiten R sollen in Verfahren 1 und 2 verwendet werden: x_1, x_2

  $x_1, x_2 \geq 0$
- Zielfunktion: Minimiere die Kosten

  $\min 3 \cdot x_1 + 5 \cdot x_2 = z \quad \Leftrightarrow \quad x_2 = -\frac{3}{5} x_1 + \frac{z}{5}$
- Nebenbedingungen: Lieferverpflichtungen

(S): $2 \cdot x_1 + 1 \cdot x_2 \geq 3 \quad \Leftrightarrow x_2 \geq -2x_1 + 3$

(M): $2 \cdot x_1 + 2 \cdot x_2 \geq 5 \quad \Leftrightarrow x_2 \geq -x_1 + \frac{5}{2}$

(L): $1 \cdot x_1 + 4 \cdot x_2 \geq 4 \quad \Leftrightarrow x_2 \geq -\frac{1}{4}x_1 + 1$

LP-Formulierung

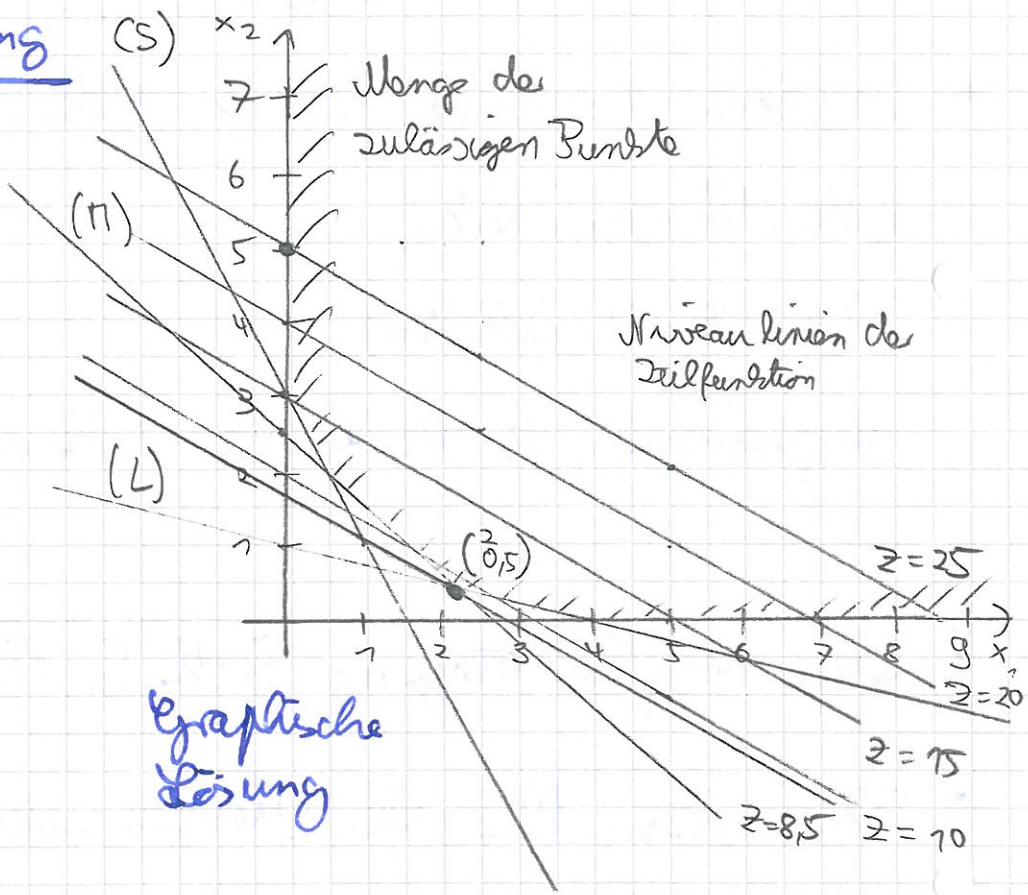
$$\min 3x_1 + 5x_2$$

$$2x_1 + x_2 \geq 3$$

$$2x_1 + 2x_2 \geq 5$$

$$x_1 + 4x_2 \geq 4$$

$$x_1, x_2 \geq 0$$



Graphische Lösung

Zulässige Punkte: Punkte, in denen alle Nebenbedingungen (einschließlich der Vorzeichenbedingungen) erfüllt sind

Niveaulinien der Zielfunktion: (genaues: Niveauhyperebenen) Mengen, auf denen die Zielfunktion einen konstanten Wert hat

Optimale Punkte: Punkte, die zulässig sind und unter allen zulässigen Punkten einen optimalen ZF-Wert haben

hier: graphisch bestimmt: $x^* = \begin{pmatrix} 2 \\ \frac{1}{2} \end{pmatrix}$ ist das einzige Minimum mit Zielfunktionswert $z^* = 8{,}5$.

Standardformen für Lineare Optimierungsprobleme

Ein LP ist in der allgemeinen Form gegeben durch eine lineare Zielfunktion

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

und lineare Nebenbedingungen als Ungleichungen und Gleichungen

$$a_{i1} x_1 + \ldots + a_{in} x_n \leq b_i \qquad i = 1, \ldots, m_I$$

$$a_{i1} x_1 + \ldots + a_{in} x_n = b_i \qquad i = m_I + 1, \ldots, m$$

Die Variablen x_i, $i = 1, \ldots, n$ können frei sein oder eine Vorzeichenbeschränkung haben:

$$x_j \geq 0 \quad , \; j = 1, \ldots, n_V$$

Matrixschreibweise:

$$\max c^T x$$

$$A_{I\cdot} x \leq b_I \qquad \text{mit } I \dot\cup E = \{1, \ldots, m\}$$

$$A_{E\cdot} x = b_E \qquad V \subset \{1, \ldots, n\}$$

$$x_V \geq 0$$

In diese Form lassen sich alle linearen Optimierungsprobleme überführen durch einfache Umformungen:

- Nebenbedingungen der Form $\sum_{j=1}^{n} a_{ij} x_j \geq b_j$ können durch Multiplikation mit -1 in $\leq$-Form gebracht werden.
- Variablen $x_i \leq 0$ können durch $-x_i \geq 0$ ersetzt werden.
- Soll die Zielfunktion minimiert werden, multipliziert man sie mit -1 und maximiert sie.

Kanonische Formen

Definition 2.1: Kanonische Formen eines LP

Sei $x \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, $A \in \mathbb{R}^{m \times n}$.

a) Die Darstellung

$$\max c^T x$$
$$Ax = b$$
$$x \geq 0$$

heißt LP in Standardform.

b) Die Darstellung

$$\max c^T x$$
$$Ax \leq b$$

heißt <u>LP in Ungleichungsform</u>.

<u>Umformungen:</u>

1.) Aus einer Gleichung können zwei Ungleichungen gemacht werden:

$$a^T x = \beta \iff \begin{cases} a^T x \leq \beta \\ -a^T x \leq -\beta \end{cases}$$

2.) Aus einer Ungleichung kann eine Gleichung und eine zusätzliche vorzeichenbeschränkte Variable gemacht werden:

$$a^T x \leq \beta \iff \begin{cases} a^T x + s = \beta \\ s \geq 0 \end{cases}$$

Die neue Variable s nennt man eine <u>Schlupfvariable</u>.

3.) Aus einer freien Variablen können zwei vorzeichenbeschränkte Variablen gemacht werden:

$$x \text{ frei} \iff \begin{cases} x = x^+ - x^- \\ x^+, x^- \geq 0 \end{cases}$$

4.) Die Vorzeichenbeschränkungen können als allgemeine Ungleichungsbeschränkungen angesehen werden:

$$x \geq 0 \iff -Ix \leq 0$$

Ebenso allgemeine untere und obere Schranken an Variablen.

Satz 2.2:

Jedes LP läßt sich in Standardform und in Ungleichungsform bringen.

Beweis:

Benutze die Umformungen. ■

Bemerkung: Vorsicht!

Umformungen von Problemen führen im allgemeinen zu einer anderen numerischen Behandlung.

Probleme sollten so modelliert werden, dass sie für die verfügbaren Methoden geeignet formuliert sind. Durch Umformungen können Strukturen verstellt werden oder schlechte Kondition entstehen.

Bemerkung:

Die Form $\min c^T x,\ Ax = b$ ist nicht universell, sie enthält keine Ungleichungen.

Die Formen $\min c^T x,\ Ax \leq b$

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

$\min c^T x,\ Ax \geq b$

sind universell.

3. Modellierung von Linearen Optimierungsproblemen, Beispielprobleme

Das Diät-Problem

Aufgabe: Stelle deine Ernährung so zusammen, dass bestimmte Anforderungen erfüllt sind und der Preis so niedrig wie möglich ist.

Beispielfall:

Food	Price	Protein	Fiber	Carbs	Calories	Cholesterol	Vitamin A	Vitamin C	Saturated fat	Sodium
Big Leo Burger	$3.29	24	3	44	530	65	10%	4%	10	1,020
Banana Split	$3.99	8	3	96	510	30	3%	33%	8	180
Raw broccoli, 1 cup	45¢	3	3	5	25	0	27%	137%	0	24
Whole grain bagel	48¢	5	3	27	140	0	0	0	0	270
2% Milk, 1 cup	40¢	8.5	0	13	130	20	10%	4%	3	125
Orange juice, 1 cup	25¢	2	1	27	110	0	2%	100%	0	0
RDA		≥ 56	≥ 30	≥ 130	∈ 1.8K, 2.2K	≤ 300	≥ 100%	≥ 100%	≤ 24	≤ 2,400

RDA: recommended daily allowance

Variablen: Welche Mengen der einzelnen Speisen soll ich täglich essen?

x_1: Burger x_2: Banana Split x_3: Broccoli

x_4: Bagel x_5: Milch x_6: Orangensaft

Zielfunktion: minimiere die Kosten in $

$$\min 3{,}25x_1 + 3{,}99x_2 + 0{,}45x_3 + 0{,}48x_4 + 0{,}40x_5 + 0{,}25x_6$$

Nebenbedingungen: Halte die erlaubten täglichen Grenzen bei verschiedenen Nährstoffe ein:

- Protein / Eiweiß:

$$24x_1 + 8x_2 + 3x_3 + 5x_4 + 8{,}5x_5 + 2x_6 \geq 50$$

- Fibe / Ballaststoffe:

$$3x_1 + 3x_2 + 3x_3 + 3x_4 + 0x_5 + 1x_6 \geq 30$$

- Carbs / Kohlenhydrate:

$$44x_1 + 96x_2 + 5x_3 + 27x_4 + 13x_5 + 27x_6 \geq 130$$

- Kalorien:

$$530x_1 + 510x_2 + 25x_3 + 140x_4 + 130x_5 + 110x_6 \geq 1800$$

$$530x_1 + 510x_2 + 25x_3 + 140x_4 + 130x_5 + 110x_6 \leq 2200$$

- Cholesterin:

$$65x_1 + 30x_2 + 0x_3 + 0x_4 + 20x_5 + 0x_6 \leq 300$$

- Vitamin A in % Tagesbedarf:

$$10x_1 + 3x_2 + 27x_3 + 0x_4 + 10x_5 + 2x_6 \geq 100$$

- Vitamin E in % Tagesbedarf:

$4x_1 + 33x_2 + 137x_3 + 0x_4 + 4x_5 + 100x_6 \geq 100$

- Fett:

$10x_1 + 8x_2 + 0x_3 + 0x_4 + 3x_5 + 0x_6 \leq 24$

- Sodium / Natrium / Salz:

$1020x_1 + 180x_2 + 24x_3 + 270x_4 + 125x_5 + 0x_6 \leq 2400$

Vorzeichenbeschränkungen: $x_1, \ldots, x_6 \geq 0$

Implementierung als LP-File: diet.lp

```
\* Problem: diet *\

Minimize
  cost: 3.29 burgers + 3.99 bananaSplit + 0.45 broccoli
        + 0.48 bagels + 0.40 milk + 0.25 orangeJuice

Subject To
   rotein:       24 burgers + 8 bananaSplit + 3 broccoli
                 + 5 bagels + 8.5 milk + 2 orangeJuice >= 56
  fiber:         3 burgers + 3 bananaSplit + 3 broccoli
                 + 3 bagels + 0 milk + 1 orangeJuice >= 30
  carbs:         44 burgers + 96 bananaSplit + 5 broccoli
                 + 27 bagels + 13 milk + 27 orangeJuice >= 130
  caloriesLb:     530 burgers + 510 bananaSplit + 25 broccoli
                 + 140 bagels + 130 milk + 110 orangeJuice <= 2200
  caloriesUb:     530 burgers + 510 bananaSplit + 25 broccoli
                 + 140 bagels + 130 milk + 110 orangeJuice >= 1800
  cholesterol:   65 burgers + 30 bananaSplit + 0 broccoli
                 + 0 bagels + 20 milk + 0 orangeJuice <= 300
  vitaminA:      10 burgers + 3 bananaSplit + 27 broccoli
                 + 0 bagels + 10 milk + 2 orangeJuice >= 100
  vitaminC:      4 burgers + 33 bananaSplit + 137 broccoli
                 + 0 bagels + 4 milk + 100 orangeJuice >= 100
  saturatedFat:  10 burgers + 8 bananaSplit + 0 broccoli
                 + 0 bagels + 3 milk + 0 orangeJuice <= 24
  sodium:        1020 burgers + 180 bananaSplit + 24 broccoli
                 + 270 bagels + 125 milk + 0 orangeJuice <= 2400

End
```

Lösung mit dem Programm glpsol

Kommandozeilenaufruf:

```
glpsol --lp diet.lp --output out.txt
```

out.txt:

```
Problem:
Rows:       10
Columns:    6
Non-zeros:  50
Status:     OPTIMAL
Objective:  cost = 5.933600425 (MINimum)

   No.   Row name   St   Activity     Lower bound   Upper bound    Marginal
------ ------------ -- ------------ ------------ ------------ ------------
     1 protein      NL           56           56                0.0234516
     2 fiber        NL           30           30                0.0683507
     3 carbs        B        388.834          130
     4 caloriesLb   B           1800                      2200
     5 caloriesUb   NL          1800         1800                0.00112635
     6 cholesterol  B        19.4772                       300
     7 vitaminA     NL           100          100                0.00542349
     8 vitaminC     B        1284.53          100
     9 saturatedFat B        2.92157                        24
    10 sodium       B        1342.18                      2400

   No. Column name  St   Activity     Lower bound   Upper bound    Marginal
------ ------------ -- ------------ ------------ ------------ ------------
     1 burgers      NL             0            0                 1.87091
     2 bananaSplit  NL             0            0                 3.00662
     3 broccoli     B        2.66482            0
     4 bagels       B        4.28332            0
     5 milk         B       0.973858            0
     6 orangeJuice  B        9.15558            0

Karush-Kuhn-Tucker optimality conditions:

KKT.PE: max.abs.err = 1.07e-14 on row 2
        max.rel.err = 1.75e-16 on row 2
        High quality

KKT.PB: max.abs.err = 0.00e+00 on row 0
        max.rel.err = 0.00e+00 on row 0
        High quality

KKT.DE: max.abs.err = 3.33e-16 on column 2
        max.rel.err = 1.12e-16 on column 2
        High quality

KKT.DB: max.abs.err = 0.00e+00 on row 0
        max.rel.err = 0.00e+00 on row 0
        High quality

End of output
```

Lösung: cost = 5,933...

burger $x_1 = 0$ banana split $x_2 = 0$

broccoli $x_3 = 2,66$ bagels $x_4 = 4,28$

milk $x_5 = 0,97$ orange juice $x_6 = 9,16$

Einige Ungleichungen sind aktiv, d.h. mit Gleichheit erfüllt.

Andere Problem:

- Orange juice ≤ 4

  cost = 6,34

  $x_1 = 0$ $x_2 = 0$ $x_3 = 2,25$ $x_4 = 6,41$ $x_5 = 3,12$ $x_6 = 4$

- Bagels ≤ 4

  cost = 7,12

  $x_1 = 0$ $x_2 = 0$ $x_3 = 4,66$ $x_4 = 4$ $x_5 = 5,25$ $x_6 = 4$

- Orange juice + Milk ≤ 6

  cost = 8,20

  $x_1 = 0,83$ $x_2 = 0$ $x_3 = 3,82$ $x_4 = 4$ $x_5 = 2$ $x_6 = 4$

- Broccoli ≤ 2

  unzulässig!

- Broccoli ≤ 3

  cost = 10,13

  $x_1 = 1{,}12 \quad x_2 = 0{,}53 \quad x_3 = 3 \quad x_4 = 4 \quad x_5 = 0 \quad x_6 = 4$

- ganz zahlig

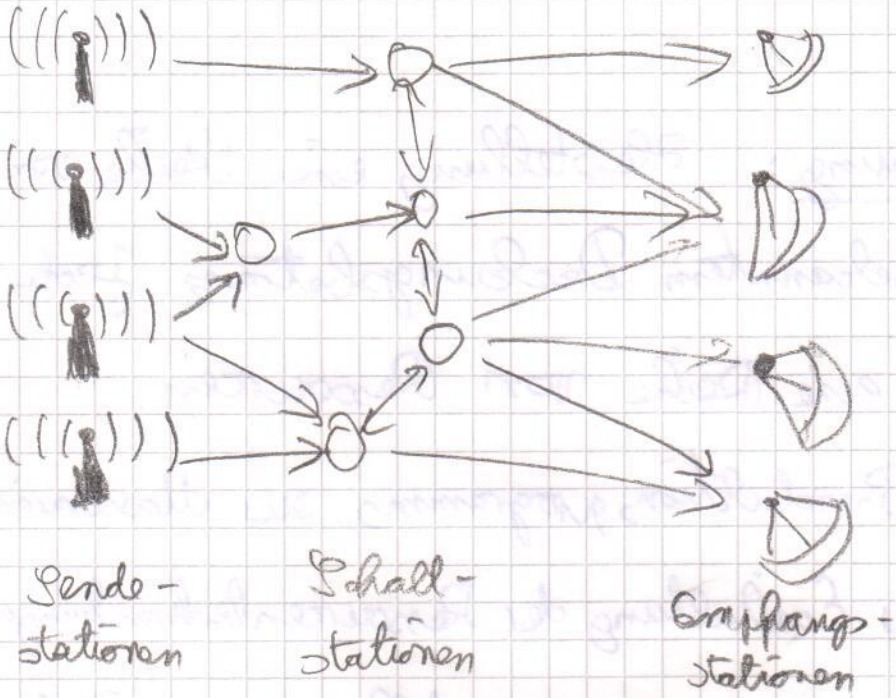
  cost = 11,30

  $x_1 = 1 \quad x_2 = 1 \quad x_3 = 3 \quad x_4 = 4 \quad x_5 = 0 \quad x_6 = 3$

Anwendung der Linearen Optimierung in der Praxis

- **Produktionsplanung:** Herstellung einer Reihe von Produkten mit bekanntem Deckungsbeitrag. Jedes Produkt benötigt eine Reihe von Ressourcen. Erstellung eines Produktionsprogramms zur Maximierung des Gewinns unter Einhaltung der Ressourcenbeschränkungen.
- **Mischungsprobleme:** Zusammenstellung von Zutaten zu einem Endprodukt. Die Menge der einzelnen Zutaten kann jeweils in einem Bereich variiert werden. Mischung von einem oder mehreren Endprodukten aus den ~~Endprodukten~~ Rohmaterialien mit minimalen Kosten.
- **Routing in Netzwerken** (z.B. Telekommunikation, Verkehr): Leiten von Verkehrsflüssen durch ein Netz, so dass alle Verkehrsanforderungen erfüllt sind, ohne die Kapazitätsbedingungen zu verletzen

Beispiel: Telekommunikationsnetzwerk



Stationen $i = 1, \dots, N$

Verbindungen $(i,j) \in \mathcal{J} \subset \{1,\dots,N\} \times \{1,\dots,N\}$

Nachrichten vom Typ $k = 1, \dots, K$

b_i^k: Anzahl der Nachrichten vom Typ k,
die in Station i gesendet werden

$b_i^k > 0$: i Sendestation

$b_i^k < 0$: i Empfangsstation

$b_i^k = 0$: i Schaltstation

c_{ij}^k : Kosten einer Nachricht vom Typ k auf Verbindung (i,j)

u_{ij} : Kapazität der Verbindung (i,j)

Ziel: Minimiere die Kommunikationskosten

Variablen x_{ij}^{k}: Anzahl der Nachrichten vom Typ k, die auf Verbindung (i,j) übertragen werden

LP-Formulierung

$$\min \sum_{k=1}^{K} \sum_{(i,j)\in J} c_{ij}^{k} x_{ij}^{k} \qquad \text{(Kosten)}$$

$$\sum_{k=1}^{K} x_{ij}^{k} \leq u_{ij} \quad \forall (i,j)\in J \qquad \text{(Kapazitätsbedingung)}$$

$$x_{ij}^{k} \geq 0$$

$$\sum_{j:(i,j)\in J} x_{ij}^{k} - \sum_{r:(r,i)\in J} x_{ri}^{k} = b_i^{k} \quad \forall i=1,\dots,N,\ k=1,\dots,K$$

(Berücksichtigung aller Nachrichten)

Typischerweise können N und J sehr groß sein.
Die Nebenbedingungen bilden Strukturen, die ausgenutzt werden können / müssen.

- Spieltheorie: Optimale Strategien in Zwei-Personen-Nullsummenspielen

- Ganzzahlige und Kombinatorische Optimierung:
  Ganzzahligkeit alle oder eines Teils der Variablen führt zu Problemen, die schwer zu lösen sind als LP's (Ganzzahlige lineare Optimierungsprobleme sind NP-schwer). Bei Lösungsverfahren dafür müssen sehr oft LP's gelöst werden (z.B. Branch & Bound).

- Nichtlineare Optimierung: Hier werden als Teilprobleme typischerweise quadratische Probleme gelöst. Die Behandlung der Ungleichungsnebenbedingungen ist dabei ähnlich wie in der linearen Optimierung.

4. Polyeder

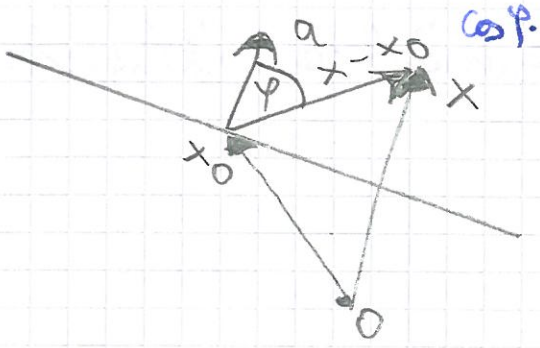
Definition 4.1:

1.) Eine Teilmenge $G_1 \subset \mathbb{R}^n$ heißt **Hyperebene**, falls es $a \in \mathbb{R}^n$, $a \neq 0$ und $\beta \in \mathbb{R}$ gibt mit
$G_1 = \{x \in \mathbb{R}^n : a^T x = \beta\}$.

2.) Eine Teilmenge $H \subset \mathbb{R}^n$ heißt **Halbraum**, falls es $a \in \mathbb{R}^n$, $a \neq 0$ und $\beta \in \mathbb{R}$ gibt mit
$H = \{x \in \mathbb{R}^n : a^T x \leq \beta\}$.

Bemerkung:

Der Normalenvektor a zeigt in den $\geq$-Halbraum.



$\cos\varphi \cdot \|a\| \|x - x_0\| = a^T(x - x_0) = a^T x - a^T x_0$
$= a^T x - \beta$

$\varphi \in [-\frac{\pi}{2}; \frac{\pi}{2}] \Rightarrow \cos\varphi \geq 0 \Rightarrow a^T x \geq \beta$

$\varphi \in [\frac{\pi}{2}; \frac{3}{2}\pi] \Rightarrow \cos\varphi \leq 0 \Rightarrow a^T x \leq \beta$

Definition 4.2:

1.) Eine Teilmenge $P \subset \mathbb{R}^n$ heißt **Polyeder**, falls es eine Matrix $A \in \mathbb{R}^{m \times n}$ und einen Vektor $b \in \mathbb{R}^n$ gibt mit

$$P = \{x \in \mathbb{R}^n : Ax \leq b\} =: P(A,b).$$

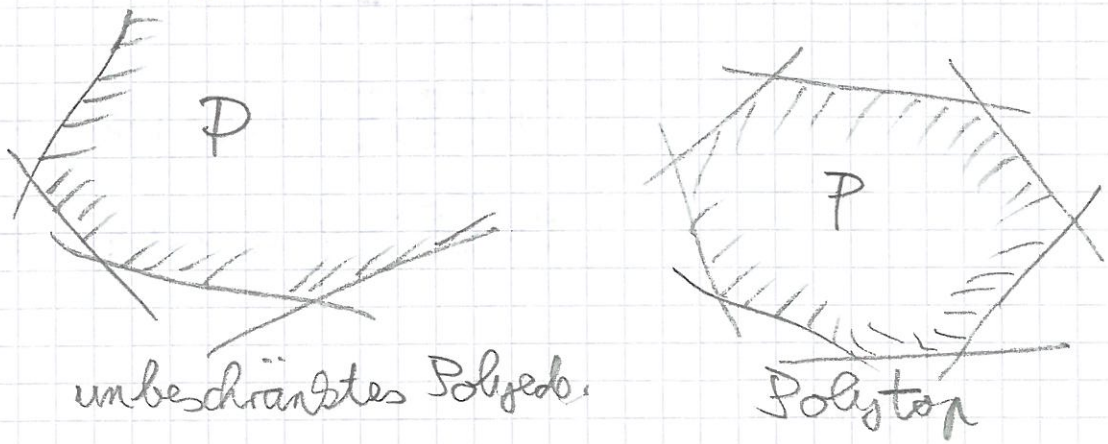
2.) Ein Polyeder P ist ein **Polytop**, wenn ein $B \in \mathbb{R}$, $B > 0$ existiert mit $P \subset \{x : \|x\| \leq B\}$.

Bemerkung:

- Ein Polyeder ist der Durchschnitt endlich vieler Halbräume:

$$\begin{matrix} a_1^T x \leq \beta_1 \\ \vdots \\ a_n^T x \leq \beta_n \end{matrix} \iff \underbrace{\begin{pmatrix} a_1^T \\ \vdots \\ a_n^T \end{pmatrix}}_{A} x \leq \underbrace{\begin{pmatrix} \beta_1 \\ \vdots \\ \beta_n \end{pmatrix}}_{b}$$

- Ein Polytop ist ein beschränktes Polyeder.



unbeschränktes Polyeder Polytop

Bemerkung:

Die Lösungsmenge des Systems

$$Bx + Cy = c$$
$$Dx + Fy \leq d$$
$$x \geq 0$$

ist ein Polyeder.

Definition 4.3: Polyedrische Kegel

Ein Kegel $C = \text{cone}(C) \subset \mathbb{R}^n$ heißt polyedrisch, wenn C ein Polyeder ist.

Satz 4.4:

Ein Kegel ist polyedrisch genau dann, wenn es eine Matrix gibt mit $C = P(A, 0)$.

Darstellungssätze:

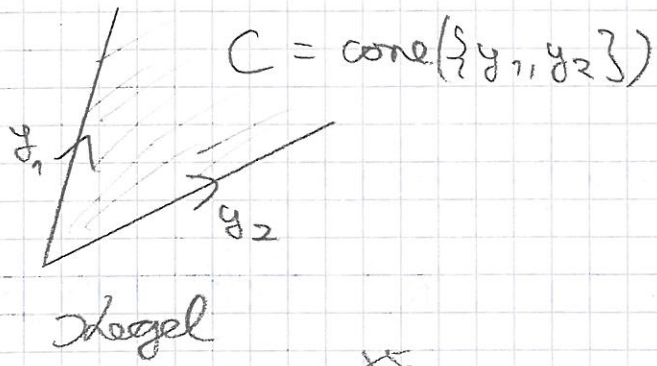
Satz 4.5:

Ein konvexer Kegel ist polyedrisch genau dann, wenn es Vektoren $y_1, \dots, y_n$ gibt mit $C = \text{cone}(\{y_1, \dots, y_n\})$.

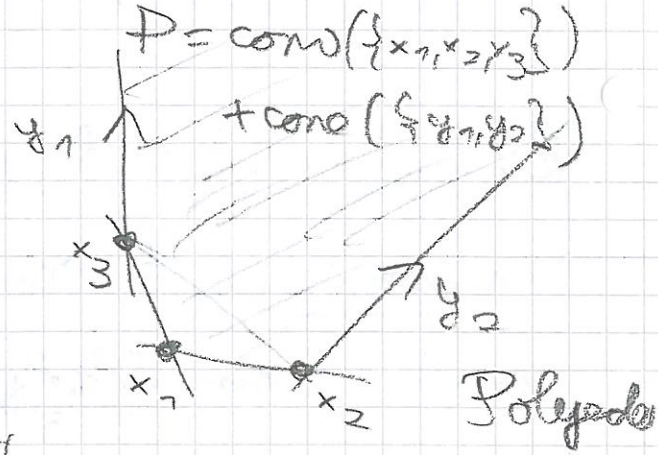
Satz 4.6:

Eine Menge P ist ein Polyeder genau dann, wenn es Vektoren $x_1, \dots, x_m$ und $y_1, \dots, y_s$ gibt mit

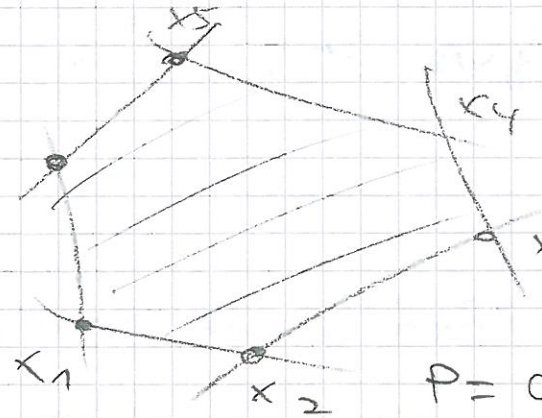
$$P = \text{conv}(\{x_1, \dots, x_m\}) + \text{cone}(\{y_1, \dots, y_s\})$$



Kegel



Polyeder



Polytop

$P = \text{conv}(\{x_1, x_2, x_3, x_4, x_5\})$

Satz 4.7:

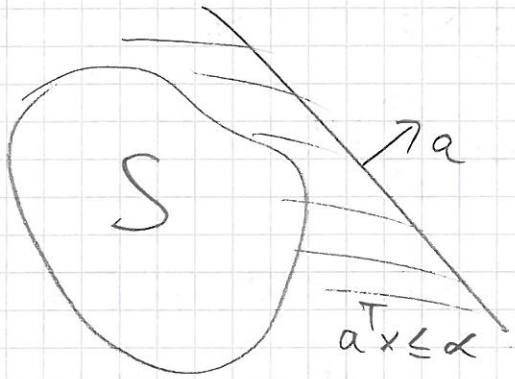
Eine Menge P ist ein Polytop genau dann, wenn es Vektoren $x_1, \dots, x_m$ gibt mit

$$P = \text{conv}(\{x_1, \dots, x_m\}).$$

Seitenflächen von Polyedern

Definition 4.8:

Seien $S \subset \mathbb{R}^n$, $a \in \mathbb{R}^n$, $\alpha \in \mathbb{R}$. Die Ungleichung $a^T x \leq \alpha$ heißt gültige Ungleichung für S, wenn $S \subset \{x \in \mathbb{R}^n : a^T x \leq \alpha\}$.



Satz 4.9: Farkas-Lemma

a) Sei $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$. Dann gilt

entweder $\exists x : Ax \leq b$ $(x \in \mathbb{R}^n)$

oder $\exists u : u^T A = 0,\ u^T b < 0,\ u \geq 0$ $(u \in \mathbb{R}^m)$

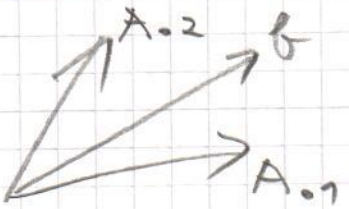
b) Sei $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$. Dann gilt

entweder $\exists x : Ax = b,\ x \geq 0$ $(x \in \mathbb{R}^n)$

oder $\exists u : u^T A \geq 0,\ u^T b < 0$ $(u \in \mathbb{R}^m)$

Ohne Beweis.

Veranschaulichung des Farkas-Lemmas:



Entweder $\exists x : Ax = b, \; x \geq 0$

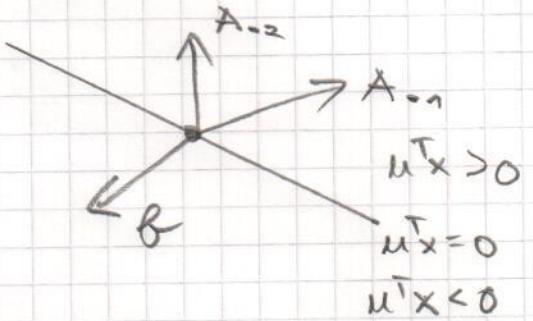
d.h. $b = \sum_{i=1}^{n} A_{\cdot i} x_i, \; x_i \geq 0$

ist konische Kombination der Spalten von A

oder $\exists u : u^T A = (u^T A_{\cdot 1}, \ldots, u^T A_{\cdot n}) \geq 0$

und $u^T b < 0$

d.h. die $A_{\cdot i}$ liegen auf einer Seite der Hyperebene $u^T x = 0$ und b auf der anderen.



Mit dem Farkas-Lemma kann man charakterisieren, ob lineare Ungleichungssysteme Lösungen haben oder nicht.

Korolla 4.10:

Seien $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$. Dann gilt:

a) Entweder $P(A,b) = \{x : Ax \leq b\} \neq \{\}$

oder $P^=(A,b) = \left\{u : \begin{pmatrix} A^T \\ b^T \end{pmatrix} u = \begin{pmatrix} 0 \\ -1 \end{pmatrix}, u \geq 0\right\} \neq \{\}$.

b) Entweder $P^=(A,b) = \{x : Ax = b, x \geq 0\} \neq \{\}$

oder $P\left(\begin{pmatrix} A^T \\ -b^T \end{pmatrix}, \begin{pmatrix} 0 \\ -1 \end{pmatrix}\right) = \left\{u : \begin{pmatrix} A^T \\ -b^T \end{pmatrix} u \leq \begin{pmatrix} 0 \\ -1 \end{pmatrix}\right\} \neq \{\}$.

Korolla 4.11:

Sei $P = P(A,b) = \{x : Ax \leq b\} = \text{conv}(V) + \text{cone}(E)$ ein Polyeder, $P \neq \{\}$. Dann sind äquivalent:

1.) $a^T x \leq \alpha$ ist eine gültige Ungleichung für P.

2.) $a^T v \leq \alpha \quad \forall v \in V$ und $a^T e \leq 0 \quad \forall e \in E$.

3.) $\exists u \geq 0 : u^T A = a^T$ und $u^T b \leq \alpha$.

Beweis 1.) $\Leftrightarrow$ 2.): Klar

1.) $\Leftrightarrow$ 3.): $a^T x \leq \alpha$ gültig für P

$\Leftrightarrow \{x : Ax \leq b,\ a^T x > \alpha\} = \{\}$

$\Leftrightarrow$ Farkas-Lemma $\{u \geq 0,\ \phi > 0 : \begin{pmatrix} u^T \\ \phi \end{pmatrix}\begin{pmatrix} A \\ -a^T \end{pmatrix} = u^T A - \phi a^T = 0,$

$\begin{pmatrix} u^T \\ \phi \end{pmatrix}\begin{pmatrix} b \\ -\alpha \end{pmatrix} = u^T b - \phi\alpha < 0\} \neq \{\}$

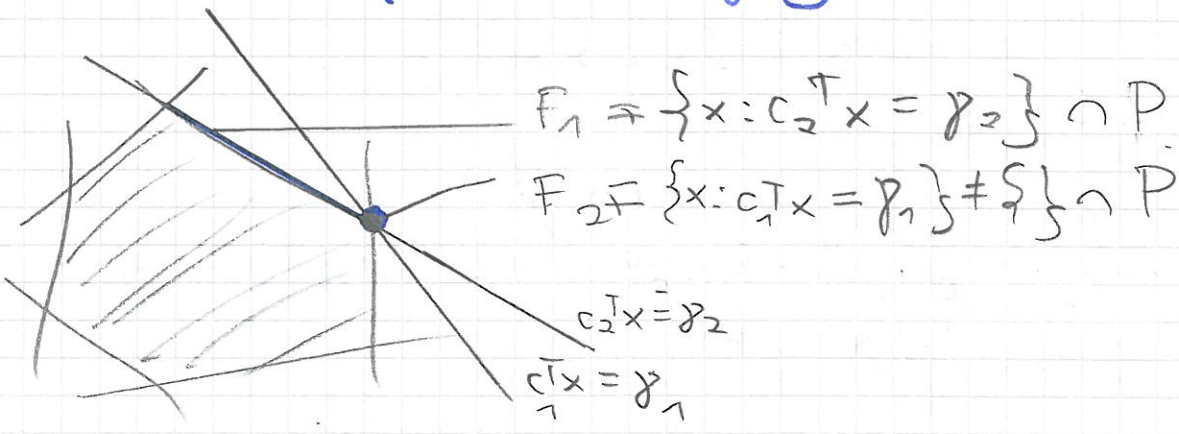
$u \in \mathbb{R}^m$, $\phi \in \mathbb{R}$

$\Leftrightarrow$ o.B.d.A. $\phi = 1$ $\{u \geq 0 : u^T A = a^T,\ u^T b < \alpha\}$

$\Leftrightarrow \exists u \geq 0 : u^T A = a^T$ und $u^T b \leq \alpha$ ∎

Definition 4.12: Seitenflächen

Sei $P \subset \mathbb{R}^n$ ein Polyeder. Eine Menge $F \subset P$ heißt **Seitenfläche** von P, wenn es eine gültige Ungleichung $c^T x \leq \gamma$ bzgl. P gibt mit $F = P \cap \{x : c^T x = \gamma\}$.



Bemerkung:

$\{\}$ und P sind immer Seitenflächen von P:

$\{\} \cap \{x : c^T x = \gamma\} \neq \{\}$ für alle c, γ.

$P \cap \{x : 0^T x = 0\} \neq \{\}$.

Alle Seitenflächen außer P heißen echte Seitenflächen.

Alle Seitenflächen außer P und $\{\}$ heißen nichttrivial

Definition 4.13:

Sei $c^T x \leq \gamma$ eine gültige Ungleichung für P.

Dann heißt $P \cap \{x : c^T x = \beta\}$ die von $c^T x \leq \beta$ induzierte Seitenfläche.

Satz 4.14:

Seien $P \subset \mathbb{R}^n$, $P = P(A, b) \neq \{\}$ und

$c \in \mathbb{R}^n$, $\gamma \in \mathbb{R}$ sowie

$$z^* := \begin{cases} +\infty, & \text{falls } c^T x \text{ auf } P \text{ unbeschränkt ist} \\ \max\{c^T x : x \in P\} & \end{cases}.$$

Dann gilt:

a) $c^T x \leq \gamma$ ist eine gültige Ungleichung für P

$\iff \gamma \geq z^*$.

b) $z^* < \infty \Leftrightarrow F = \{x \in P : c^T x = z^*\}$
ist nicht leere Seitenfläche von P.

Beweis:

a) $x \in P \Rightarrow c^T x \leq z^* \Rightarrow$ Für alle $\gamma \geq z^*$
gilt $c^T x \leq \gamma$ für alle $x \in P$.
$\gamma \geq \max\{c^T x, x \in P\} \Rightarrow c^T x \leq \gamma \quad \forall x \in P$.

b) $z^* < \infty \Leftrightarrow \exists x^* \in P : z^* = c^T x^* \Leftrightarrow F \cap P \neq \{\}$. ■

Korollar 4.15:

Die Menge der Optimallösungen eines LP ist eine Seitenfläche des durch die Nebenbedingungen definierten Polyeders.

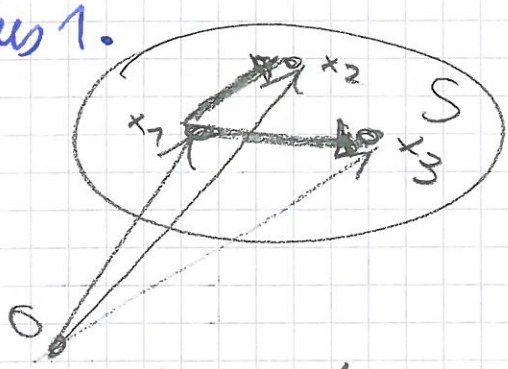
Satz 4.16:

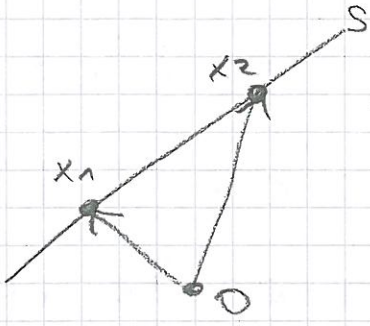
Sei $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$. Dann hat $P(A, b)$ höchstens $2^m + 1$ Seitenflächen.

Beweis: In einer Seitenfläche ist jede Nebenbedingung entweder mit Gleichheit erfüllt oder nicht. Zu zeigt: ■

Definition 4.17: Dimension einer Menge

Die Dimension einer Menge S ist die maximale Anzahl von affin unabhängigen Punkten in S minus 1.





$\dim S = 1$

$v_1 = x_2 - x_1$ und

$v_2 = x_3 - x_1$ sind lin. abh.,

deshalb sind x_1, x_2, x_3 affin unabh.

$\dim S = 2$

Definition 4.18: Sei P ein Polyeder.

Seitenflächen der Dimension 0:

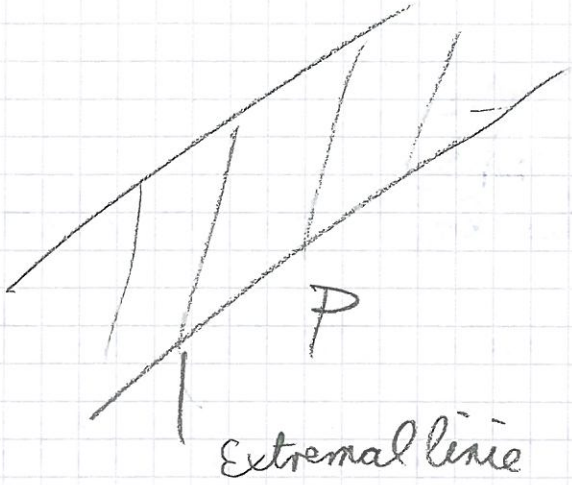
$F = \{x\}$ heißen Ecken von P.

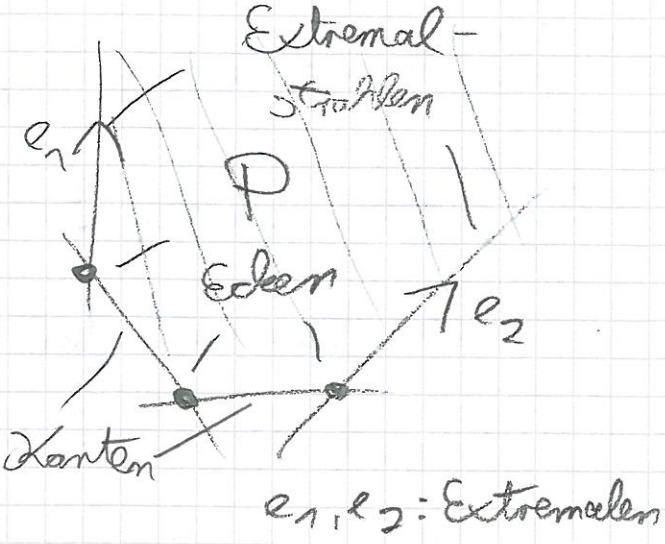
Seitenflächen der Dimension 1, speziell

$F = x + \text{lin}(\{z\})$ heißen Extremallinien,

$F = x + \text{cone}(\{z\})$ heißen Extremalstrahlen,

$F = \text{conv}(\{x,y\})$, x, y Ecken, heißen Kanten.





Definition 4.19:
Ein Polyeder heißt spitz, wenn es eine Ecke besitzt.

Satz 4.20:
Sei C ein polyedrische Kegel. Dann gilt:
x ist Ecke von C $\Rightarrow x = 0$.

Definition 4.21:
Die Richtungsvektoren der Extremalstrahlen von P heißen Extremalen von P.

Satz 4.22: Darstellungssatz
1.) Jeder spitze polyedrische Kegel $C \neq \{0\}$ ist die Summe seiner Extremalstrahlen:

$C = \text{cone}(\{e_1, \dots, e_m\})$

2.) Jedes spitze Polyeder ($P \neq \{\}$) läßt sich darstellen als die Summe der konvexen Hülle seiner Ecken und der konischen Hülle seiner Extremalen:

$$P = \text{conv}(V) + \text{cone}(E)$$
$$= \text{conv}(\underbrace{\{v_1, \dots, v_m\}}_{\text{Ecken}}) + \text{cone}(\underbrace{\{e_1, \dots, e_{\ell'}\}}_{\text{Extremalen}})$$

3.) Jedes Polyton ($P \neq \{\}$) läßt sich darstellen als die konvexe Hülle seiner Ecken.

$$P = \text{conv}(V) = \text{conv}(\{v_1, \dots, v_m\}).$$

Definition 4.23:

Ein Element x eines Polyeders P heißt innerer Punkt, wenn x in keiner echten Seitenfläche von P enthalten ist.

Satz 4.24:

Jedes nicht leere Polyeder besitzt innere Punkte.

Definition 4.25: Facetten

Eine nichttriviale Seitenfläche von $P(A,b)$ heißt Facette von $P(A,b)$, falls sie in keiner anderen echten Seitenfläche von $P(A,b)$ enthalten ist.

Satz 4.26: Sei P ein volldimensionales Polyeder: $\dim P = n$.
$F = \{x \in P : c^T x = \gamma\}$ ist Facette von $P(A,b)$ genau dann, wenn $\dim F = n-1$.

Satz 4.27: Charakterisierung von Ecken

Sei $P = \{x \in \mathbb{R}^n : Ax \leq b\} = P(A,b)$ und $x \in P$.

Dann sind äquivalent:

a) x ist Ecke von P.

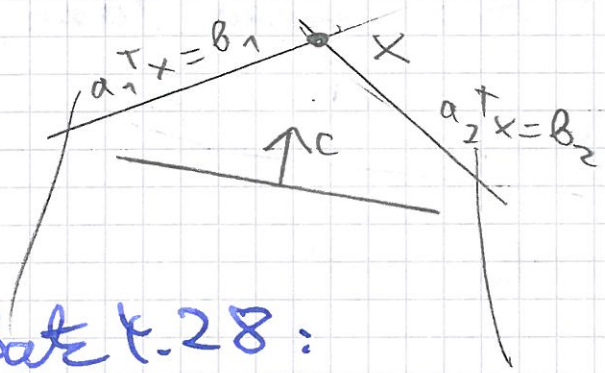
b) $\{x\}$ ist eine 0-dimensionale Seitenfläche von P.

c) x ist keine echte Konvexkombination von Elementen

aus P. (d.h. $\nexists\, y,z \in P,\ y \neq z,\ 0<\lambda<1$:
$x = \lambda y + (1-\lambda) z$)

d) Es gibt eine vollrangige Teilmatrix
$A_{I\cdot}$ von A, so dass $A_{I\cdot} x = b_I$
(Rang $A_{I\cdot} = n$).

e) Es gibt ein $c \in \mathbb{R}^n - \{0\}$, so dass x eindeutige
Optimallösung von $\max \{c^T y, y \in P\}$ ist.



Satz 4.28:

a) Enthält ein nichtleeres Polyeder keine Gerade, so hat es eine Ecke.

b) Ist $P = \{x \in \mathbb{R}^n : Ax = b, x \geq 0\}$ nicht leer,
so hat P eine Ecke.

c) Hat ein Polyeder P eine Ecke, so auch jede
nichtleere Seitenfläche von P.

d) Ein nichtleeres Polytop hat eine Ecke.

Satz 4.29:

$\max\{c^T x : Ax \leq b, x \geq 0\}$ bzw $\max\{c^T x : Ax = b, x \geq 0\}$ besitzen optimale Ecklösungen, wenn sie optimale Lösungen besitzen.

Beweis:
Nach Satz 4.14 ist die Menge aller Optimallösungen eine Seitenfläche. Nach Satz 4.28 haben die zulässigen Polyeder der zulässigen Mengen eine Ecke, also auch jede ihrer Seitenflächen. Die Ecke der optimalen Seitenfläche ist eine optimale Ecklösung. ∎

Satz 4.30:

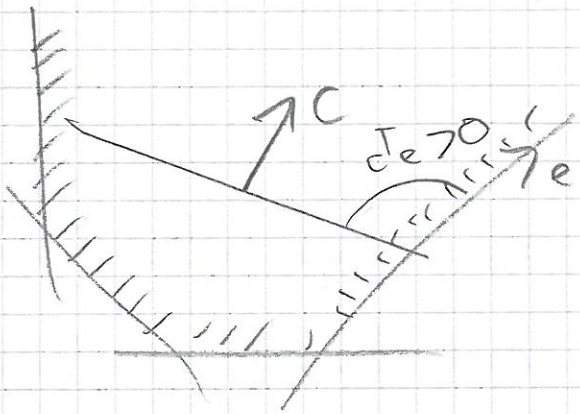
Sei $P = P(A, b)$ ein nichtleeres, spitzes Polyeder. Dann ist $\max\{c^T x, x \in P\}$ genau dann unbeschränkt, wenn es eine Extremale e von P gibt mit $c^T e > 0$.

Beweis:

$\Leftarrow$: $x \in P$, $x + \lambda e \in P \quad \forall \lambda > 0$

$\Rightarrow c^T(x + \lambda e) = c^T x + \lambda \underbrace{c^T e}_{>0}$ unbeschränkt.

$\Rightarrow$: Benutze Farkas-Lemma.



Zielfunktion ist entlang Extremale unbeschränkt.

5. Dualitätstheorie

$$(P)\quad \max\ c^T x \quad \Leftrightarrow \quad \max\ z$$
$$Ax \le b \qquad\qquad z - c^T x \le 0$$
$$Ax \le b$$

Gesucht:

x^* mit $Ax^* \le b$

und $c^T x^* \ge c^T x$

für alle $x \in P(A,b)$

Für festes z: $P_z := \left\{ x \in \mathbb{R}^n : \begin{pmatrix} -c^T \\ A \end{pmatrix} x \le \begin{pmatrix} -z \\ b \end{pmatrix} \right\}$

Optimierungsaufgabe:

$$\max\{ z : P_z \neq \{\} \}$$

primal: Nachweis, dass eine Menge nicht leer ist

dual: Nachweis, dass eine Menge leer ist

Obere Schranke: $\min\{ z : P_z = \{\} \}$

$$\Downarrow$$

$$\max\{ z : P_z \neq \{\} \}$$

Nach Farkas-Lemma ist $P_z = \{\}$ genau dann, wenn es λ, y gibt, so dass

$$\begin{pmatrix} \lambda \\ y \end{pmatrix}^T \begin{pmatrix} -c^T \\ A \end{pmatrix} = -\lambda c^T + y^T A \quad = 0$$

$$\begin{pmatrix} \lambda \\ y \end{pmatrix}^T \begin{pmatrix} -z \\ b \end{pmatrix} = -\lambda z + y^T b \quad < 0$$

$$y \geq 0 \qquad \lambda \geq 0$$

Falls Lösung mit $\lambda = 0$ existiert, ist $P_z = \{\}$ für alle z.

Wenn (P) eine Optimallösung hat, gibt es eine Lösung des Systems mit $\lambda > 0$.

Sei $(\bar{\lambda}, \bar{y})$ eine solche Lösung, dann ist auch $\frac{1}{\bar{\lambda}}(\bar{\lambda}, \bar{y}) = (1, y)$ eine Lösung des Systems:

$$y^T A = c^T$$
$$y^T b < z$$
$$y \geq 0$$

Bestimme das kleinste z:

$$\begin{aligned} &\min\ y^T b \\ &y^T A = c^T \qquad (D) \\ &y \geq 0 \end{aligned}$$

Satz 5.1: Schwacher Dualitätssatz

Seien $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$.

Seien (P) und (D) die beiden folgenden linearen Programme:

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

Dann gilt: Für Punkte $x_0 \in \mathbb{R}^n$ und $y_0 \in \mathbb{R}^m$ mit $Ax_0 \leq b$, $y_0^T A = c^T$, $y_0 \geq 0$ gilt:

$$c^T x_0 \leq y_0^T b.$$

Beweis:

$$c^T x_0 = y_0^T A x_0 \leq y_0^T b \qquad \blacksquare$$

Satz 5.2: Dualitätssatz

Seien $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$.

Die beiden linearen Programme

(P) $\max c^T x$ und (D) $\min y^T b$

$Ax \leq b$ $\quad$ $y^T A = c^T$

$y \geq 0$

haben optimale Lösungen mit dem gleichen Zielfunktionswert genau dann, wenn beide zulässige Lösungen haben.

Beweis:

Seien $\bar{x}$ und $\bar{y}$ zulässige Lösungen.

Nach dem schwachen Dualitätssatz ist $c^T \bar{x} \leq \bar{y}^T b$.

Die beiden Probleme haben dann Lösungen mit dem gleichen Zielfunktionswert, wenn das folgende System eine Lösung hat:

$Ax \leq b$ $\quad$ $b^T y - c^T x \leq 0$ $\quad$ $(\Leftrightarrow c^T x \geq b^T y)$

$A^T y = c$ $\quad$ $y \geq 0$

bzw.

$$\left.\begin{aligned} \begin{pmatrix} A \\ -c^T \end{pmatrix} x + \begin{pmatrix} 0 \\ e^T \end{pmatrix} y &\leq \begin{pmatrix} b \\ 0 \end{pmatrix} \\ A^T y &= c \\ y &\geq 0 \end{aligned}\right\} (*)$$

Lösbarkeit dieses Systems ist nach dem Farkas-Lemma äquivalent zur Nichtlösbarkeit von

$$\left.\begin{aligned} u^T A - z c^T &= 0 \\ z b^T + v^T A^T &\geq 0 \\ u \geq 0 \quad z &\geq 0 \\ u^T b + v^T c &< 0 \end{aligned}\right\} (**)$$

1. Fall: Es existiert Lösung mit $z = 0$

$\Rightarrow P = \{\}$ oder $D = \{\}$ ↯

2. Fall: Es existiert Lösung mit $z > 0$

$\Rightarrow \exists$ Lösung mit $z = 1$

$\Rightarrow 0 > u^T b + v^T c \geq u^T(-Av) + v^T(A^T u) = 0$ ↯

Also ist (**) nicht lösbar und damit (*) lösbar.

■

Korollar 5.3:

P und D seien die Lösungsmengen von (P) und (D).

Wir setzen

$$z^* := \begin{cases} +\infty & \text{, wenn (P) unbeschränkt} \\ -\infty & \text{, wenn } P = \{\} \\ \max\{c^T x : x \in P\} & \text{sonst} \end{cases}$$

und

$$u^* := \begin{cases} -\infty & \text{, wenn (D) unbeschränkt} \\ +\infty & \text{, wenn } D = \{\} \\ \min\{u^T b : u \in D\} & \text{sonst} \end{cases}$$

Dann gilt:

a) z^* endlich $\Leftrightarrow$ u^* endlich $\Leftrightarrow -\infty < z^* = u^* < +\infty$

b) $z^* = +\infty \Rightarrow D = \{\}$

c) $u^* = -\infty \Rightarrow P = \{\}$

d) $D = \{\} \Rightarrow P = \{\}$ oder $z^* = +\infty$

e) $P = \{\} \Rightarrow D = \{\}$ oder $u^* = -\infty$

Beweis:

Zu zeigen bleibt a): (P) hat Optimallösung $\Rightarrow D \neq \{\}$

Sei z^* endlich, d.h. für $z_0 > z^*$ ist das System

$-c^T x \leq -z_0$ $\quad Ax \leq b$ unlösbar.

Nach dem Farkas-Lemma ist dann

$-u c^T + y^T A = 0$, $-u z_0 + y^T b < 0$, $u, y \geq 0$

lösbar. Sei $\bar{u}, \bar{y}$ Lösung.

Falls $\bar{u} = 0$. $\Rightarrow$ $P = \{\} \not\subseteq$

Falls $\bar{u} > 0$ $\Rightarrow$ System ist lösbar mit $\bar{u} = 1$.

$\Rightarrow$ Es gibt ein $\bar{y}$ mit $\bar{y}^T A = c^T$, $\bar{y}^T b < z_0$, $\bar{y} \geq 0$

$\Rightarrow$ $\bar{y} \in D$, d.h. $D \neq \{\}$

b), c), d) und e) folgen aus a) und dem Dualitätssatz. ■

Beispiel: Beide Lösungsmengen leer

$$P = \left\{ x : \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} x \leq \begin{pmatrix} 0 \\ -1 \end{pmatrix} \right\} = \{\}$$

$$D = \left\{ y : \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} y = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, y \geq 0 \right\} = \{\}$$

Die linearen Programme

$\max (1,1)\, x$ und $\min (0,-1)\, y$

$x \in P$ $\qquad$ $y \in D$

sind dual zueinander, $P = \{\}$, $D = \{\}$.

Definition 5.4 : Primales und duales Problem

Gegeben seien dimensionsverträgliche Matrizen A, B, C, D und Vektoren a, b, c, d.

Das zum primalen Problem

$$\max c^T x + d^T y$$

$$\begin{aligned} Ax + By &\leq a \\ Cx + Dy &= b \\ x &\geq 0 \end{aligned} \qquad (P)$$

zugehörige duale Problem ist

$$\min u^T a + v^T b$$

$$\begin{aligned} u^T A + v^T C &\geq c^T \\ u^T B + v^T D &= d^T \\ u &\geq 0. \end{aligned} \qquad (D)$$

Beispiele / Spezialfälle:

primal	dual
1.) $\max d^T y$	$\min u^T a$
$B y \leq a$	$u^T B = d^T$
	$u \geq 0$
2.) $\max c^T x$	$\min v^T b$
$Cx = b$	$v^T C \geq c^T$
$x \geq 0$	
3.) $\max c^T x$	$\min u^T a$
$Ax \leq a$	$u^T A \geq c^T$
$x \geq 0$	$u \geq 0$

usw.

Da die Fälle 1.) und 2.) (z.B.) sich in jedes allgemeine LP transformieren lassen, folgt der allgemeine Fall aus Definition 5.4 schon aus den Spezialfällen 1.) und 2.), siehe Übungsaufgabe.

Faustregeln:

Gleichung → Variable, nicht vorzeichenbeschränkt

Ungleichung ($\leq$) → Variable, vorzeichenbeschränkt (≥ 0)

Variable, vorzeichenbeschränkt (≥ 0) → Ungleichung (≤ 0)

Variable, nicht vorzeichenbeschränkt → Gleichung

rechte Seite → Zielfunktion

Zielfunktion → rechte Seite

max → min

min → max

Satz 5.5:

Das zum dualen Problem gehörende duale Problem ist das primale Problem.

Beweis: Übungsaufgabe.

Satz 5.6: Satz vom schwachen komplementären Schlupf

Sei (P) und (D) ein Paar zueinander duale Probleme.
Die Vektoren $\begin{pmatrix}\bar{x}\\ \bar{y}\end{pmatrix}$ und $\begin{pmatrix}\bar{u}\\ \bar{v}\end{pmatrix}$ seien zulässig für (P) bzw (D).
Dann sind äquivalent:

(a) $\begin{pmatrix}\bar{x}\\ \bar{y}\end{pmatrix}$ ist optimal für (P) und $\begin{pmatrix}\bar{u}\\ \bar{v}\end{pmatrix}$ ist optimal für (D).

(b) $\left(c^T - (\bar{u}^T A + \bar{v}^T C)\right)\bar{x} - \bar{u}^T(a - (A\bar{x} + B\bar{y})) = 0$

(c) Für alle Komponenten $\bar{u}_i$ gilt:

$\bar{u}_i > 0 \Rightarrow A_{i\cdot}\bar{x} + B_{i\cdot}\bar{y} = a_i$

Für alle Komponenten $\bar{x}_j$ gilt:

$\bar{x}_j > 0 \Rightarrow \bar{u}^T A_{\cdot j} + \bar{v}^T C_{\cdot j} = c_j$

(d) $A_{i\cdot}\bar{x} + B_{i\cdot}\bar{y} < a_i \Rightarrow \bar{u}_i = 0$

$\bar{u}^T A_{\cdot j} + \bar{v}^T C_{\cdot j} > c_j \Rightarrow \bar{x}_j = 0$

Beweis: Benutze den Dualitätssatz.

Der Satz besagt also: Wenn eine Dualvariable > 0 ist, ist die zugehörige Ungleichung des primalen Problems mit Gleichheit erfüllt. (die Ungleichung ist aktiv)

Beispiel: Orangenproblem

$$\max\ 0{,}2x_1 + 0{,}25x_2 \qquad \max\ c^T x$$

$$2x_1 + x_2 \le 200$$

$$0{,}2x_1 + x_2 \le 110 \qquad Ax \le b \qquad (P)$$

$$x_1, x_2 \ge 0 \qquad x \ge 0$$

(D) $\min\ u^T b$ $\qquad \min\ 200u_1 + 110u_2$

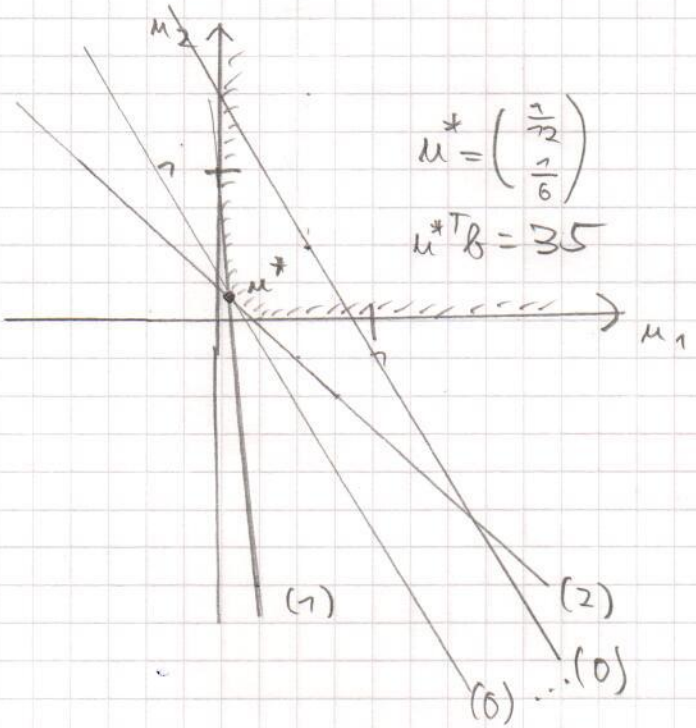
$$u^T A \ge c^T \qquad (u_1, u_2)\begin{pmatrix} 2 & 1 \\ 0{,}2 & 1 \end{pmatrix} \ge (0{,}2 \;\; 0{,}25)$$

$$u \ge 0 \qquad u_1, u_2 \ge 0$$

$$\min\ 200u_1 + 110u_2 = z \iff u_1 = \frac{z}{110} - \frac{20}{11}u_1 \quad (0)$$

$$2u_1 + 0{,}2u_2 \ge 0{,}2 \iff u_2 \ge 1 - 10u_1 \quad (1)$$

$$u_1 + u_2 \ge 0{,}25 \iff u_2 \ge 0{,}25 - u_2 \quad (2)$$



In der Lösung:

$u_1 > 0$, $u_2 > 0$

d.h. im Maximum von (P) x^* gelt:

$$2x_1^* + x_2^* = 200$$

$$0{,}2x_1^* + x_2^* = 110$$

$$\Rightarrow (x_1^*, x_2^*) = (50, 100)$$

Und wenn eine Primalvariable > 0 ist, ist die zugehörige Ungleichung des dualen Problems mit Gleichheit erfüllt. (Ungleichung aktiv)

Nach diesem Satz könnte noch gelten, dass Variable und zugehörige Ungleichung mit Gleichheit erfüllt sind (Ungleichung schwach aktiv). Dies schließt aber der folgende Satz aus.

Satz 5.7: Satz vom starken Komplementärschlupf

Besitzen die beiden linearen Programme

(P) $\max c^T x \quad Ax \leq b$ und

(D) $\min u^T b \quad u^T A = c^T \quad u \geq 0$

Lösungen, dann existieren Optimallösungen $\bar{x}$ und $\bar{u}$, so dass für alle Zeilen in dieses i von A gilt:

$\bar{u}_i > 0 \Leftrightarrow A_{i\cdot}\bar{x} = b_i$

$A_{i\cdot}x < b \Leftrightarrow \bar{u}_i = 0$

Beweis: Benutze Farkas-Lemma.

Beispiel: Orangenproblem

$\max\ 0{,}2x_1 + 0{,}25x_2$ $\qquad \max\ c^T x$

$2x_1 + x_2 \leq 200$

$0{,}2x_1 + x_2 \leq 110$ $\qquad Ax \leq b \qquad (P)$

$x_1, x_2 \geq 0$ $\qquad x \geq 0$

(D) $\min\ u^T b$ $\qquad \min\ 200u_1 + 110u_2$

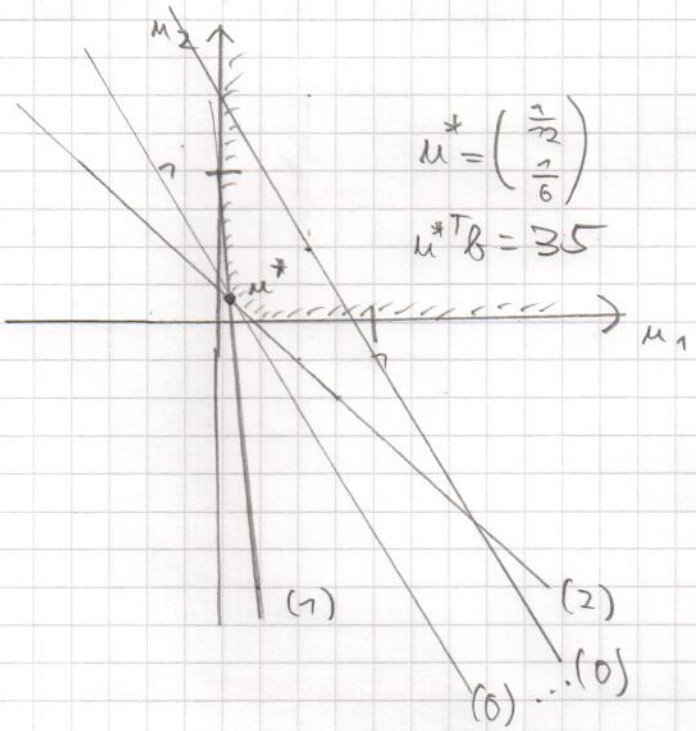
$u^T A \geq c^T$ $\qquad (u_1, u_2)\begin{pmatrix} 2 & 1 \\ 0{,}2 & 1 \end{pmatrix} \geq (0{,}2 \;\; 0{,}25)$

$u \geq 0$ $\qquad u_1, u_2 \geq 0$

$\min\ 200u_1 + 110u_2 = z \quad \Leftrightarrow \quad u_1 = \frac{z}{110} - \frac{20}{11}u_1 \ (0)$

$2u_1 + 0{,}2u_2 \geq 0{,}2 \quad \Leftrightarrow \quad u_2 \geq 1 - 10u_1 \ (1)$

$u_1 + u_2 \geq 0{,}25 \quad \Leftrightarrow \quad u_2 \geq 0{,}25 - u_2 \ (2)$



$u^* = \begin{pmatrix} \frac{1}{12} \\ \frac{1}{6} \end{pmatrix}$

$u^{*T} b = 35$

In der Lösung:

$u_1 > 0$, $u_2 > 0$

d.h. im Maximum von (P) x^* gelt:

$2x_1^* + x_2^* = 200$

$0{,}2x_1^* + x_2^* = 110$

$\Rightarrow (x_1^*, x_2^*) = (50, 100)$

6. Die Simplex-Methode (geometrische Variante)

Sei $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$.

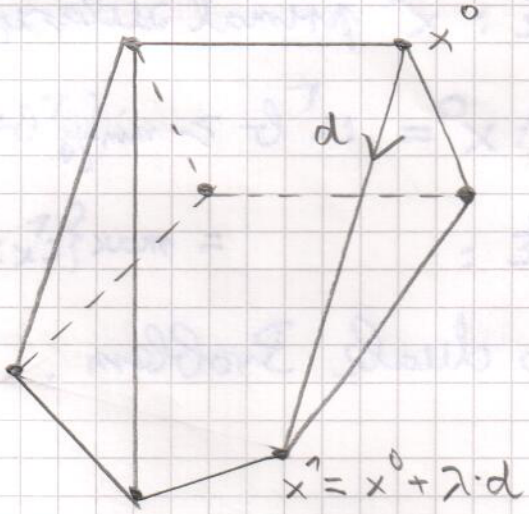
Lineares Programm:

$$\max c^T x$$
$$Ax \leq b$$

Duales Programm:

$$\min u^T b$$
$$u^T A = c^T, \quad u \geq 0$$

Annahme: $P(A,b) = \{x \in \mathbb{R}^n : Ax \leq b\}$ besitzt Ecke.
Ecke x^0 sei bekannt.



Es gibt eine reguläre $n \times n$-Teilmatrix von A

mit $A_B \cdot x^0 = b_B$ $\quad B$: Zeilenindexmenge der Teilmatrix

$A_N \cdot x^0 \leq b_B$ $\quad N = \{1, \dots, m\} \setminus B$

a) Ist x^0 optimale Eckenlösung?

Definiere Vektor $u=(u_B, u_N)$ durch

$u_B^T := c^T A_B^{-1}$, $u_N^T := 0$.

Es gilt dann $u^T A = u_B^T A_{B\cdot} + u_{N\cdot}^T = c^T$

Falls $u_B \geq 0$: Für alle $y \in P(A,b)$ gilt dann:

$c^T y = u_B^T A_{B\cdot} y \leq u_B^T b_B = u_B^T A_{B\cdot} x^0 = c^T x^0$

$\Rightarrow x^0$ ist Optimallösung.

Schwache Dualitätssatz: x^0 primal zulässig, u dual zulässig: $c^T x^0 = u^T A x^0 = u^T b \geq \min\{y^T b : y \geq 0, y^T A = c^T\}$

Starker Dualitätssatz: $= \max\{c^T x : Ax \leq b\} = c^T x^0$

$\Rightarrow u$ optimal für das duale Problem.

STOP.

β) Bestimmung einer Anstiegsrichtung d

Sei $i \in B$ ein Index mit $u_i < 0$.

$d := -A_{B\cdot}^{-1} e_i$ (also d ist i-te Spalte von $-A_{B\cdot}^{-1}$)

$$\Rightarrow c^T d = -c^T A_{B\cdot}^{-1} e_i = -u_B^T e_i = -u_i > 0$$

$$c^T(x^0 + \lambda d) - c^T x^0 = \lambda c^T d \geq 0 \quad \forall \lambda \geq 0 .$$

c) Bestimmung der Schrittweite $\lambda \geq 0$

Gesucht: $\lambda \geq 0$, so dass $x^0 + \lambda d \in P(A, b)$

$$A_{j\cdot}(x^0 + \lambda d) = A_{j\cdot} x^0 + \lambda A_{j\cdot} d \overset{!}{\leq} b_j \quad \forall j = \{1, \dots, m\}$$

$j \in B \setminus \{i\}$: $A_{j\cdot} x^0 = b_j$, $A_{j\cdot} d = -e_j^T e_i = 0$

$j = i$: $A_{j\cdot} x^0 = b_i$, $A_{j\cdot} d = -e_i^T e_i = -1$

$j \in N$: $A_{j\cdot} x^0 \leq b_j$

Falls (i) $A_{j\cdot} d \leq 0$, ist jedes $\lambda \geq 0$ zulässig

Falls (ii) $A_{j\cdot} d > 0$, muss gelten:

$$A_{j\cdot}(x^0 + \lambda d) \leq b_j \Leftrightarrow \lambda \leq \frac{b_j - A_{j\cdot} x^0}{A_{j\cdot} d}$$

Also: Falls $A_{j\cdot} d \leq 0 \ \forall j \in N$ $(\Leftrightarrow Ad \leq 0)$

ist d Extremale von $P(A, b)$ mit $c^T d > 0$

$\Rightarrow$ Problem ist unbeschränkt. STOP.

d) Übergang zur neuen Ecke x^1

$$\lambda := \min\left\{ \frac{b_j - A_j \cdot x^0}{A_j \cdot d} \,,\ j \in N,\ A_j \cdot d > 0 \right\} \geq 0$$

$x^1 := x^0 + \lambda d$ Dann ist $x^1 \in P(A, b)$.

Sei $j \in N$ ein Index mit $\lambda = \frac{b_j - A_j \cdot x^0}{A_j \cdot d}$

$$A_k \cdot x^1 = A_k \cdot x^0 + \lambda A_k \cdot d = \begin{cases} b_k + \lambda \cdot 0 \,, & k \in B \setminus \{i\} \\ A_j \cdot x^0 + b_j - A_j \cdot x^0 \,, & k = j \end{cases} = b_k$$

$\forall\, k \in (B \setminus \{i\}) \cup \{j\}$

$\Rightarrow x^1$ Ecke

Ersetze B durch $(B \setminus \{i\}) \cup \{j\}$

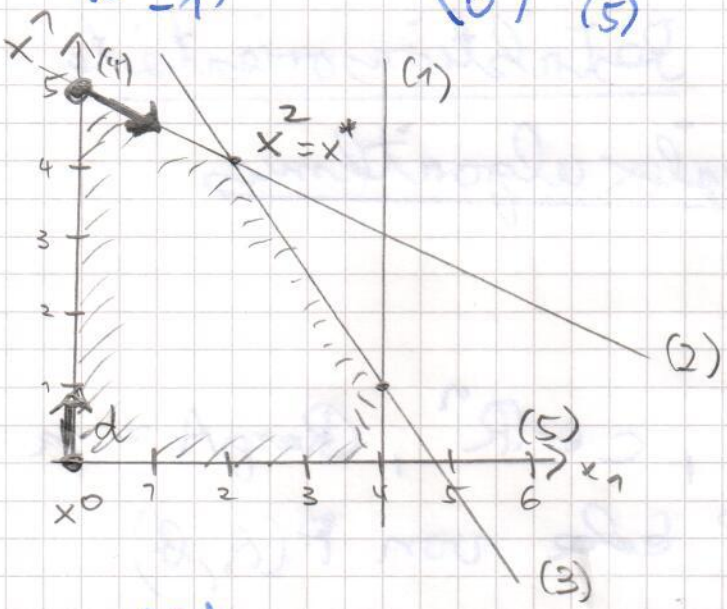
und x^0 durch x^1.

Weiter bei a).

Beispiel:

$$\max (1,1)\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 0 \\ 1 & 2 \\ 3 & 2 \\ -1 & 0 \\ 0 & -1 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \leq \begin{pmatrix} 4 \\ 10 \\ 14 \\ 0 \\ 0 \end{pmatrix} \begin{matrix} (1) \\ (2) \\ (3) \\ (4) \\ (5) \end{matrix}$$



$$x^0 = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad B = \{4,5\} \quad A_{B\cdot} = \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix} = A_{B\cdot}^{-1}$$

a) $u_B^T = c^T A_{B\cdot}^{-1} = (-1\ -1) \quad u_N^T = (0,0,0)$

nicht optimal

b) Wähle $i = 5$, $d := -A_B^{-1} \cdot e_i = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$

c)

$j \in N$	$A_{j\cdot} d$	$b_j - A_{j\cdot} x^0$	λ
1	0	–	
2	2	10	5
3	2	14	7

d) $\bar{j} = 2 \quad \lambda = 5 \quad x^1 = x^0 + \lambda d = \begin{pmatrix} 0 \\ 0 \end{pmatrix} + 5\begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 5 \end{pmatrix}$

Weiter im Beispiel:

$x^1 = \begin{pmatrix} 0 \\ 5 \end{pmatrix}$ $\quad B = \{2,4\}$ $\quad A_{B\cdot} = \begin{pmatrix} 1 & 2 \\ -1 & 0 \end{pmatrix}$ $\quad A_{B\cdot}^{-1} = \begin{pmatrix} 0 & -1 \\ \frac{1}{2} & \frac{1}{2} \end{pmatrix}$

a) $u_B^T = c^T A_{B\cdot}^{-1} = \begin{pmatrix} \frac{1}{2} & -\frac{1}{2} \end{pmatrix}$ $\quad u_N^T = (0,0,0)$

nicht optimal

b) $i = 4$, $d := -A_B^{-1} \cdot e_i = \begin{pmatrix} 1 \\ -\frac{1}{2} \end{pmatrix}$

c)

$j \in N$	$A_{j\cdot} d$	$b_j - A_{j\cdot} x^1$	λ
1	1	4	4
3	2	4	2
5	0,5	5	10

d) $j = 3$ $\quad \lambda = 2$

$x^2 = x^1 + \lambda d = \begin{pmatrix} 2 \\ 4 \end{pmatrix}$ $\quad B = \{2,3\}$

$A_{B\cdot} = \begin{pmatrix} 1 & 2 \\ 3 & 2 \end{pmatrix}$ $\quad A_{B\cdot}^{-1} = \begin{pmatrix} -\frac{1}{2} & \frac{1}{2} \\ \frac{3}{4} & -\frac{1}{4} \end{pmatrix}$

a) $u_B^T = c^T A_{B\cdot}^{-1} = \begin{pmatrix} \frac{1}{4} & \frac{1}{4} \end{pmatrix}$ $\quad u_N^T = (0,0,0)$

optimal

Optimallösung: $x^* = \begin{pmatrix} 2 \\ 4 \end{pmatrix}$ mit ZF-Wert 6

und Dualvariable $u^{*T} = (0, \frac{1}{4}, 0, \frac{1}{4}, 0)$

Algorithmus 6.1: Restriktionsorientierte Variante des Simplexalgorithmus

Input:

$A \in \mathbb{R}^{n \times n}$, $b \in \mathbb{R}^n$, $c \in \mathbb{R}^n$, $\operatorname{Rang} A = n$

$x^0 \in \mathbb{R}^n$ mit x^0 Ecke von $P(A, b)$

Teilsystem $A_0 x \leq b_0$ von $Ax \leq b$ mit $A_0 x^0 = b_0$ und A_0 regulär.

Output:

Entweder x^* mit $c^T x^* = \max\{c^T x : Ax \leq b\}$ oder Nachweis der Unbeschränktheit des Problems.

(0) Setze $i := 0$ und bestimme A_i^{-1} (bzw. Zerlegung)

(1) Bestimme u so, dass $c^T = u^T A$
und $u_j = 0$ für Komponenten j nicht im Teilsystem A_i
(durch Berechnung von $c^T A_i^{-1}$).

(2) Falls $u \geq 0$, dann setze $x^* = x^i$.
STOP: x^* ist Optimallösung.

(3) Sei i^* der kleinste Index einer negativen Komponente von u und d die zugehörige Spalte von $-A_i^{-1}$ ($d := -A_i^{-1} e_{i^*}$).

(4) Falls $Ad \leq 0$:
STOP: Problem ist unbeschränkt, da d Extremale mit $c^T d > 0$.

(5) Bestimme $\lambda_i := \min \left\{ \frac{b_j - A_{j\cdot} x^i}{A_{j\cdot} d}, j \in \{1, \dots, m\}, A_{j\cdot} d > 0 \right\}$

(6) Sei j^* der kleinste Index mit $\lambda_i = \frac{b_{j^*} - A_{j^*\cdot} x^i}{A_{j^*\cdot} d}$

(7) Setze $x^{i+1} := x^i + \lambda_i d$.
Bilde neues Teilsystem A_{i+1} (ersetze i^* durch j^*).

Bestimme A_{i+1}^{-1} (bzw. Zerlegung), setze $i := i+1$, gehe zu (1).

Satz 6.2:

Der Simplex-Algorithmus 6.1 terminiert in endlich vielen Schritten.

Beweisskizze:

- Der Algorithmus produziert eine Folge von Iterationspunkten mit nicht fallenden Zielfunktionswerten.
- Durch die Auswahlregeln „kleinster Index" in den Schritten (3) und (6) wird erreicht, dass kein Teilsystem zweimal vorkommt, d.h. es gibt keine Iterationen $k < \ell$ mit $A_k = A_\ell$ und $x^k = x^\ell$.
- Es gibt nur endlich viele Teilsysteme, die alle höchstens einmal vorkommen.

$\Rightarrow$ Der Algorithmus terminiert. ∎

Wie bestimmt man eine zulässige Startlösung?

Durch Lösen eines sogenannten Phase-I-Hilfsproblems.

Für $\max c^T x$ $\quad (P)$ $\quad Ax \leq b$ betrachte das

Hilfsproblem:

$$\max \lambda$$
$$Ax - b\lambda \leq 0$$
$$-\lambda \leq 0$$
$$\lambda \leq 1$$

Dieses Problem ist ein LP in der für Alg 6.1 geeigneten Form.

$x = 0, \lambda = 0$ ist eine zulässige Ecke.

Löse das Hilfsproblem mit Alg. 6.1: **Phase I.**

Die Lösung $(\bar{x}, \bar{\lambda})$ des Hilfsproblems erfüllt

- entweder $\bar{\lambda} = 1 \Rightarrow A\bar{x} - b \leq 0$
  $\Rightarrow \bar{x}$ ist zulässig für (P)
  Löse (P) dann mit Alg 6.1 und Startlösung $\bar{x}$: **Phase II.**

- oder $\bar{\lambda} < 1 \Rightarrow$ (P) unzulässig.

Satz 6.3: Übergang zur neuen Ecke

Sei x^0 zul. Ecke, d.h. $Ax^0 \leq b$, $A_{B^0} \cdot x^0 = b_{B^0}$, A_{B^0} vollrangig. Sei $i \in B^0$ Index mit $u_i = c^T A_{B^0}^{-1} e_i < 0$, $d := -A_{B^0}^{-1} e_i$.

Sei $\lambda := \min \left\{ \frac{b_j - A_j \cdot x^0}{A_j \cdot d}, j \notin B^0, A_j \cdot d > 0 \right\} \geq 0$.

Sei $j \in B^0$ ein Index mit $A_j \cdot d > 0$, $\lambda = \frac{b_j - A_j \cdot x^0}{A_j \cdot d}$ und $x^1 := x^0 + \lambda d$.

Dann gilt:

(i) x^1 ist zulässig, d.h. $Ax^1 \leq b$.

(ii) x^1 ist Ecke, d.h. $A_{B^1} \cdot x^1 = b_{B^1}$ mit $B^1 = (B^0 \setminus \{i\}) \cup \{j\}$, A_{B^1} vollrangig.

(iii) $c^T x^1 \geq c^T x^0$.

Beweis:

(i) $A_{k\cdot} x^1 = A_{k\cdot} x^0 + \lambda A_{k\cdot} d$

(a) $= b_k + \lambda \cdot 0 = b_k$ für $k \in B^0 \setminus \{i\}$

(b) $= A_{j\cdot} x^0 + \frac{b_j - A_{j\cdot} x^0}{A_{j\cdot} d} A_{j\cdot} d = b_j$ für $k = j$

(c) $= b_i + \lambda \cdot (-1)$ für $k = i$

(d) $\leq b_k + \lambda A_{k\cdot} d \leq b_k$ für $k \notin B^0$, $A_{k\cdot} d \leq 0$

(e) $= A_{k\cdot} x^0 + \lambda A_{k\cdot} d \leq b_k$ für $k \notin B^0$, $A_{k\cdot} d > 0$,

da $\lambda = \frac{b_j - A_{j\cdot} x^0}{A_{j\cdot} d} \leq \frac{b_k - A_{k\cdot} x^0}{A_{k\cdot} d}$

$\Rightarrow A x^1 \leq b$, x^1 zulässig und

(ii) $A_{B^1\cdot} x^1 = b_{B^1}$ Rang $A_{B^1\cdot} \leq n$

Zeige Rang $A_{B^1\cdot} = n$

Es gilt $A_{k\cdot} d = -e_k^T A_{B^0\cdot} A_{B^0\cdot}^{-1} e_i = -e_k^T e_i = 0$

für $k \in B^0 - \{i\}$ und $A_{j\cdot} d > 0$.

n.V. Rang $A_{B^0\cdot} = n$,

d.h. $\sum_{k \in B^0} \mu_k A_{k\cdot} = 0 \Rightarrow \mu_k = 0 \quad \forall k \in B^0$

insbes. $\sum_{k \in B^0 \setminus \{i\}} \mu_k A_{k\cdot} = 0 \Rightarrow \mu_k = 0 \quad \forall k \in B^0 \setminus \{i\}$

$$\sum_{\ell \in B^1} \mu_\ell A_{\ell\cdot} = \sum_{\ell \in B^0 \setminus \{i\}} \mu_\ell A_{\ell\cdot} + \mu_j A_{j\cdot} = 0$$

$$\Rightarrow 0 = \sum_{\ell \in B^0 \setminus \{i\}} \mu_\ell A_{\ell\cdot} d + \mu_j A_{j\cdot} d$$

$$= 0 + \mu_j A_{j\cdot} d \Rightarrow \mu_j = 0$$

$$\Rightarrow \sum_{\ell \in B^0 \setminus \{i\}} \mu_\ell A_{\ell\cdot} = 0 \Rightarrow \mu_\ell = 0 \quad \forall \ell \in B^0 \setminus \{i\}$$

$\Rightarrow A_{\ell\cdot}$, $\ell \in B^1$ linear unabhängig

$\Rightarrow$ Rang $A_{B^1\cdot} = n \Rightarrow x^1$ Ecke.

(iii) $c^T x^1 = c^T (x^0 + \lambda d) = c^T x^0 + \lambda c^T d$

$= c^T x^0 \underbrace{-\lambda}_{\geq 0} \underbrace{c^T A_{B^0\cdot}^{-1} e_i}_{= \mu_i < 0} \geq c^T x^0$ ■

7. Die Grundversion der Simplex-Methode

Betrachte nun lineare Programme in Standardform

$$\max c^T x$$
$$Ax = b$$
$$x \geq 0$$

mit $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$

Voraussetzungen:

(a) A ist $m \times n$-Matrix mit $m < n$.

(b) Rang $A = m$

(c) $\{x : Ax = b, x \geq 0\} \neq \{\} \Rightarrow$ hat Ecken

Bezeichnungen:

Zeilenindexmenge von A: $\{1, \dots, m\}$

Spaltenindexmenge von A: $\{1, \dots, n\}$

B und N bezeichnen Spaltenindexvektoren von A

$B = (\lambda_1, \dots, \lambda_m) \in \{1, \dots, n\}^m$

$N = (\varrho_1, \dots, \varrho_{n-m}) \in \{1, \dots, n\}^m$

Kein Index kommt doppelt oder in beiden Vektoren vor.

B und N können auch als Mengen aufgefasst werden.

$i \in B$, $j \in N$.

$B \cup N = \{1, \dots, n\}$, $B \cap N = \{\}$

$A_B := A_{\cdot B}$ $A_N := A_{\cdot N}$ Teilmatrizen

Mit B ist N bestimmt und es ist $N = (q_1, \dots, q_{n-m})$ mit $q_i < q_j$ für $i < j$.

Definition 7.1. (Basislösung)

Gegeben sei das Gleichungssystem $Ax = b$ und die Spaltenindexvektoren B und N.

a) Ist A_B regulär, so heißt A_B Basismatrix (Basis) von A, A_N heißt Nichtbasismatrix (Nichtbasis) von A.

b) Ist A_B Basis, so heißt der Vektor $x^T = (x_B^T, x_N^T)$ Basislösung (von $Ax = b$ zur Basis B), falls $x_B = A_B^{-1} b$, $x_N = 0$.

c) $x_j, j \in B$ heißt Basisvariable,

$x_j, j \in N$ heißt Nichtbasisvariable.

d) Eine Basislösung x heißt zulässig,

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

e) Eine Basislösung x heißt nichtdegeneriert,

falls $x_B > 0$, andernfalls degeneriert.

Satz 7.2:

Sei $P := \{x : Ax = b, x \geq 0\} \subset \mathbb{R}^n$ ein Polyeder,

Rang $(A) = m < n$ und $x \in P$.

Der Punkt x ist Ecke von P genau dann,

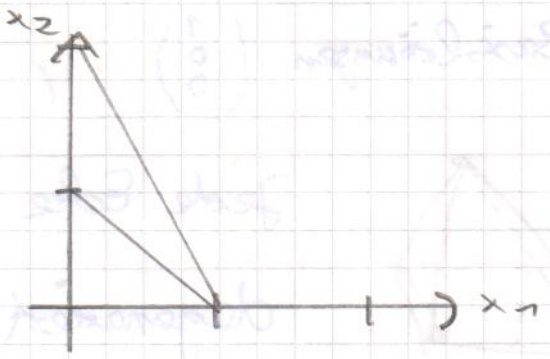
wenn x eine zulässige Basislösung ist.

Beispiele:

a) $x_1 + x_2 \leq 1$

$2x_1 + x_2 \leq 2$

$x_1, x_2 \geq 0$



$\Rightarrow x_1 + x_2 + s_1 = 1$

$2x_1 + x_2 + s_2 = 2$

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

$x_1, x_2, s_1, s_2 \geq 0$ $\quad G = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$

$A_B = \begin{pmatrix} 1 & 1 \\ 2 & 1 \end{pmatrix}$ (z. B.) $\quad B = (1,2) \quad N = (3,4)$

$A_B^{-1} = \begin{pmatrix} -1 & 1 \\ 2 & -1 \end{pmatrix}$ $\quad x_B = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \begin{matrix} x_1 \\ x_2 \end{matrix}$

$A_B = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$ $\quad B = (2,3) \quad N = (1,4)$

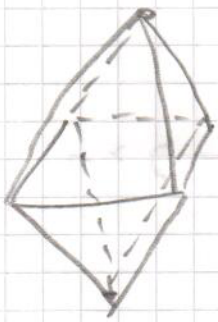
$A_B^{-1} = \begin{pmatrix} 0 & 1 \\ 1 & -\frac{1}{2} \end{pmatrix}$ $\quad x_B = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \begin{matrix} x_1 \\ s_1 \end{matrix}$

b) $A = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$ $\quad b = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$

zwei Basen $\begin{pmatrix} 1 & 0 \\ 0 & 1 \\ x_1 & x_3 \end{pmatrix}$, $\begin{pmatrix} 1 & 0 \\ 0 & 1 \\ x_2 & x_3 \end{pmatrix}$

zug. Basislösungen $\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$, $\begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$

c)



Jede Ecke ist degeneriert

Basistausch

Satz 7.3:

Sei das System $Ay = b$ mit Rang $A = m$ gegeben. A_B sei Basis von A. Dann gilt für beliebiges x: $Ax = b$ genau dann, wenn

$$x_B = A_B^{-1} b - A_B^{-1} A_N x_N.$$

Beweis:

$$Ax = b \Leftrightarrow (A_B\ A_N)\begin{pmatrix} x_B \\ x_N \end{pmatrix} = b$$

$$\Leftrightarrow A_B^{-1}(A_B\ A_N)\begin{pmatrix} x_B \\ x_N \end{pmatrix} = A_B^{-1} b$$

$$\Leftrightarrow x_B + A_B^{-1} A_N x_N = A_B^{-1} b \quad \blacksquare$$

Satz 7.4:

Gegeben sei ein Gleichungssystem $Ay = b$, $B = (\gamma_1, \dots, \gamma_m)$ und $N = (\varrho_1, \dots, \varrho_{n-m})$ seien die Spaltenindexvektoren zur Basis A_B.

Definiere $\bar{A} := A_B^{-1} A_N = (\bar{a}_{rs})_{\substack{r \in B \\ s \in N}}$

$$\bar{b} := A_B^{-1} b = (\bar{b}_r)_{r \in B}.$$

a) Ist $\bar{a}_{rs} \neq 0$, so ist ein <u>Basistausch</u> möglich:

$A_{B'}$ mit $B' = (\lambda_1, \dots, \lambda_{r-1}, q_s, \lambda_{r+1}, \dots, \lambda_m)$ ist eine Basis von A ($\bar{a}_{rs}$ heißt <u>Pivotelement</u>).

b) Es gilt: $A_{B'}^{-1} = E A_B^{-1}$, wobei E die folgende Matrix ist:

$$E = \begin{pmatrix} 1 & & \eta_1 & & \\ & \ddots & \vdots & & \\ & & \eta_{r-1} & & \\ & & \eta_r & & \\ & & \eta_{r+1} & 1 & \\ & & \vdots & & \ddots \\ & & \eta_m & & 1 \end{pmatrix}$$

$$\eta_r = \frac{1}{\bar{a}_{rs}}$$

$$\eta_i = -\frac{\bar{a}_{is}}{\bar{a}_{rs}} \quad i \neq r$$

η: <u>Eta-Vektor</u>

c) Sind x und x' die zu A_B und $A_{B'}$ gehörenden Basislösungen, so gilt:

$$x'_{\lambda_i} = \bar{b}_i - \frac{\bar{a}_{is}}{\bar{a}_{rs}} \bar{b}_r = E_{i\cdot} \bar{b} = E_{i\cdot} x_B$$

für $i \in \{1, \dots, m\} - \{r\}$

$$x'_{q_s} = \frac{1}{\bar{a}_{rs}} \bar{b}_r = E_{r\cdot} \bar{b} = E_{r\cdot} x_B$$

$$x'_j = 0 \text{ sonst}$$

Beweis: Nachrechnen.

Beispiel:

$$A = \begin{pmatrix} 1 & 0 & 2 & -\frac{3}{2} & -4 & \frac{11}{2} & \frac{11}{2} \\ 0 & 1 & 0 & 0 & 3 & 0 & 2 \\ 0 & 0 & -1 & \frac{1}{2} & 2 & -\frac{3}{2} & -\frac{1}{2} \\ 0 & 0 & 0 & \frac{1}{2} & 1 & -\frac{1}{2} & -\frac{1}{2} \end{pmatrix} \quad b = \begin{pmatrix} 1 \\ 2 \\ -1 \\ 2 \end{pmatrix}$$

$B = (1,2,3,4) \quad N = (5,6,7)$

$$A_B^{-1} b = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \end{pmatrix} \quad x^T = (1,2,3,4,0,0,0)$$

Basistausch: $2 \leftrightarrow 7 \quad B' = (1,7,3,4)$

$$E = \begin{pmatrix} 1 & -2 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & \frac{1}{2} & 0 & 1 \end{pmatrix}$$

$$A_{B'}^{-1} b = \begin{pmatrix} -3 \\ 1 \\ 3 \\ 5 \end{pmatrix} \quad x'^T = (-3,0,3,5,0,0,1)$$

Satz 7.5:

Gegeben sei ein lineares Programm in Standardform mit Basis A_B.

a) Für den Zielfunktionswert $c^T x$ eines $x \in P = \{x : Ax = b, x \geq 0\}$ gilt:

$$c^T x = c_B^T A_B^{-1} b + (c_N^T - c_B^T A_B^{-1} A_N) x_N .$$

(Der Term $\bar{c}^T := c_N^T - c_B^T A_B^{-1} A_N$ heißt reduzierte Kosten).

b) Ist A_B eine zulässige Basis und sind die reduzierten Kosten nicht positiv, dann ist die zugehörige Basislösung $x_B = A_B^{-1} b$, $x_N = 0$ optimal.

c) Ist A_B eine zulässige Basis und ist die zugehörige Basislösung nichtdegeneriert und optimal, dann sind die reduzierten Kosten nicht positiv.

Beweis:

a) $c^T x = \begin{pmatrix} c_B^T & c_N^T \end{pmatrix} \begin{pmatrix} x_B \\ x_N \end{pmatrix}$

Einsetzen von $x_B = A_B^{-1} b - A_B^{-1} A_N x_N$:

$$c^T x = c_B^T A_B^{-1} b + \left(c_N^T - c_B^T A_B^{-1} A_N \right) x_N$$

b) Sei $y \in P$ beliebig.

$$c^T y = c_B^T A_B^{-1} b + \underbrace{\bar{c}^T}_{\leq 0} \underbrace{y_N}_{\geq 0} \leq c_B^T A_B^{-1} b = c_B^T x_B = c^T x$$

c) Annahme: Es gibt q_j mit $\bar{c}_j > 0$.

x nichtdegeneriert $\Rightarrow A_B^{-1} b > 0$.

Dann gibt es ein $\lambda > 0$ mit

$$A_B^{-1} b \geq A_B^{-1} A_N \lambda e_j$$

x^λ mit $x_B^\lambda = A_B^{-1} b - A_B^{-1} A_N \lambda e_j$, $x_N^\lambda = \lambda e_j$

ist zulässig: $A x^\lambda = b - A_N \lambda e_j + A_N \lambda e_j = b$

und $x_B^\lambda \geq 0$, $x_N^\lambda \geq 0$, und es gilt:

$$c^T x^\lambda = c_B^T A_B^{-1} b + \underbrace{\lambda \bar{c}^T e_j}_{>0} > c_B^T A_B^{-1} b = c^T x \quad \lightning$$

Satz 7.6:

Gegeben sei ein LP in Standardform.

A_B sei zulässige Basis mit Basislösung x.

Sei $\bar{A} := A_B^{-1} A_N$, $\bar{b} := A_B^{-1} b$, $\bar{c}^T := c_N^T - c_B^T A_B^{-1} A_N$

und $q_j \in N$ ein Index mit $\bar{c}_j > 0$. Dann gelten:

a) Ist $\bar{A}_{\cdot j} \leq 0$, dann ist $c^T x$ auf $P = \{x : Ax = b, x \geq 0\}$ unbeschränkt.

b) Ist $\bar{A}_{\cdot j} \neq 0$, dann setzen wir

$$\lambda_0 := \min \left\{ \frac{\bar{b}_i}{\bar{a}_{ij}}, i = 1, \dots, m, \bar{a}_{ij} > 0 \right\}$$

und wählen $r \in \{1, \dots, m\}$ mit $\frac{\bar{b}_r}{\bar{a}_{rj}} = \lambda_0$, $\bar{a}_{rj} > 0$.

Dann ist $A_{B'}$ mit $B' = (\pi_1, \dots, \pi_{r-1}, q_j, \pi_{r+1}, \dots, \pi_m)$ eine neue Basis mit Lösung x' und $c^T x' \geq c^T x$.
Ist A_B nicht degeneriert, dann ist $c^T x' > c^T x$.

Beweis:

a) Sei $y \in P$ und $\lambda > 0$. Dann ist $y_B = A_B^{-1} b - A_B^{-1} A_N y_N \geq 0$, $y_N \geq 0$. Setze $y'_N := y_N + \lambda e_j \geq 0$. Dann ist

$$y'_B = A_B^{-1} b - A_B^{-1} A_N y_N - \lambda \bar{A}_{\cdot j} = y_B \underbrace{- \lambda \bar{A}_{\cdot j}}_{\geq 0 \text{ n.V.}} \geq 0.$$

$$c^T y' = \underbrace{c_B^T A_B^{-1} b + \bar{c}^T y_N}_{= c^T y} + \underbrace{\lambda \bar{c}_j}_{> 0} > c^T y$$

$\Rightarrow$ Zielfunktion wird beliebig groß.

b) Nach Satz 7.4 ist $B' = (\pi_1, \dots, \pi_{r-1}, q_j, \pi_{r+1}, \dots, \pi_m)$ eine Basis mit Basislösung x': $x'_{\pi_i} = \bar{b}_i - \frac{\bar{a}_{ij}}{\bar{a}_{rj}} \bar{b}_r \geq 0$, $i \neq r$

($\bar{b}_r \geq 0$, $\bar{a}_{rj} > 0$, $\bar{b}_i \geq 0$ und $\frac{\bar{b}_r}{\bar{a}_{rj}} \leq \frac{\bar{b}_i}{\bar{a}_{ij}}$ für $\bar{a}_{ij} > 0$)

und $x'_{q_j} = \frac{\bar{b}_r}{\bar{a}_{rj}} \geq 0$, $x'_j = 0$ sonst $\Rightarrow x'$ zulässig.

$$c^T x' = c_B^T A_B^{-1} b + \left(c_N^T - c_B^T A_B^{-1} A_N\right) x'_N$$

$$= \underbrace{c_B^T A_B^{-1} b}_{= c^T x} + \underbrace{\bar{c}_j}_{> 0} \underbrace{\frac{\bar{b}_r}{\bar{a}_{rj}}}_{\geq 0} \geq c^T x$$

Nichtdegeneriertheit: $\bar{b}_r > 0 \Rightarrow \frac{\bar{b}_r}{\bar{a}_{rj}} > 0 \Rightarrow c^T x' > c^T x$ ■

Algorithmus 7.7: Grundversion der Simplexmethode

Input: A', b', c

Output: Optimale Lösung von $\max\{c^T x : A'x = b', x \geq 0\}$
oder Nachweis der Unzulässigkeit
oder Nachweis der Unbeschränktheit

Phase I:

(I.1) Bestimme ein Subsystem $Ax = b, x \geq 0$
mit $\{x : Ax = b, x \geq 0\} = \{x : A'x = b', x \geq 0\}$
und A hat vollen Zeilenrang m. Sei $m < n$.

(I.2) Bestimme $B = (\pi_1, \dots, \pi_m)$, so dass A_B
zulässige Basis ist. Sei $N := (\varrho_1, \dots, \varrho_{n-m})$.

(I.3) Berechne A_B^{-1}, $\bar{A} = A_B^{-1} A_N$, $\bar{b} = A_B^{-1} b$
$\bar{c}^T = c_N^T - c_B^T A_B^{-1} A_N$, $c_0 = c_B^T \bar{b}$.

Phase II:

(II.1) Gilt $\bar{c} \leq 0$, so ist die aktuelle Basislösung
optimal. x mit $x_B = \bar{b}$, $x_N = 0$ ist optimal
mit Zielfunktionswert c_0. STOP.

(II.2) (Pivotspaltenauswahl)

Wähle einen Index $s \in \{1, \dots, n-m\}$ mit $\bar{c}_s > 0$.

(II.3) Falls $\bar{A}_{\cdot s} \leq 0$, so ist das LP unbeschränkt. STOP.

(II.4) (Bestimmung der Schrittweite)

Berechne $\lambda_0 := \min \left\{ \frac{\bar{b}_i}{\bar{a}_{is}}, \bar{a}_{is} > 0, i = 1, \dots, m \right\}$

(II.5) (Pivotzeilenauswahl)

Wähle einen Index $r \in \{1, \dots, m\}$ mit $\lambda_0 = \frac{\bar{b}_r}{\bar{a}_{rs}}$

(II.6) Setze $B' := (\lambda_1, \dots, \lambda_{r-1}, q_s, \lambda_{r+1}, \dots, \lambda_m)$,

$N' := (q_1, \dots, q_{s-1}, \lambda_r, q_{s+1}, \dots, q_{n-m})$

$A_{B'}^{-1} := E A_B^{-1}$.

(II.7) a) Neuberechnung von $\tilde{A} = A_{B'}^{-1} A_{N'} = E A_B^{-1} A_{N'}$

$\tilde{a}_{is} := -\frac{\bar{a}_{is}}{\bar{a}_{rs}}, \quad i = 1, \dots, m, \ i \neq r$

$\tilde{a}_{rj} := \frac{\bar{a}_{rj}}{\bar{a}_{rs}}, \quad j = 1, \dots, n-m, \ j \neq s$

$\tilde{a}_{ij} := \bar{a}_{ij} - \frac{\bar{a}_{is}}{\bar{a}_{rs}} \bar{a}_{rj}$ sonst

b) Neuberechnung von $\tilde{b} = E \bar{b}$:

$\tilde{b}_r := \frac{\bar{b}_r}{\bar{a}_{rs}}$

$\tilde{b}_i := \bar{b}_i - \frac{\bar{a}_{is}}{\bar{a}_{rs}} \bar{b}_r \quad i = 1, \dots, m, \ i \neq r$

c) Neuberechnung von $\tilde{c}$ und $\tilde{c}_0$:

$$\tilde{c}_j := -\frac{\bar{c}_j}{\bar{a}_{rj}}$$

$$\tilde{c}_\partial := \bar{c}_\partial - \frac{\bar{a}_{r\partial}}{\bar{a}_{rj}}\bar{c}_j \,, \quad \partial = 1,\dots,n-m, \ \partial \neq j$$

$$\tilde{c}_0 := c_0 + \bar{c}_j \frac{\bar{b}_r}{\bar{a}_{rj}}$$

Setze $\bar{A} := \tilde{A}$, $\bar{b} := \tilde{b}$, $\bar{c} := \tilde{c}$, $c_0 := \tilde{c}_0$

und gehe zu (II.1)

Ende.

$$Ax = b \iff A_B x_B + A_N x_N = b$$

$$\iff A_B^{-1} b = A_B^{-1} A \begin{pmatrix} x_B \\ x_N \end{pmatrix}$$

$$c_0 = c^T x \iff c_0 = c_B^T x_B + c_N^T x_N$$

$$\iff c_0 = c_B^T A_B^{-1} b + \left(c^T - c_B^T A_B^{-1} A\right)\begin{pmatrix} x_B \\ x_N \end{pmatrix}$$

$$\iff -c_B^T A_B^{-1} b = \left(c^T - c_B^T A_B^{-1} A\right)\begin{pmatrix} x_B \\ x_N \end{pmatrix} - c_0$$

Definition 7.8:

Sei $\max\{c^T x : Ax = b, x \geq 0\}$ ein LP in Standardform mit Basis A_B.

a) Die $(m+1) \times (n+1)$ - Matrix

$$\left(\begin{array}{c|c} -c_B^T A_B^{-1} b & c^T - c_B^T A_B^{-1} A \\ \hline A_B^{-1} b & A_B^{-1} A \end{array}\right)$$

heißt Simplex - Tableau zur Basis B.

b) Die $(m+1) \times (n-m+1)$ - Matrix

$$\left(\begin{array}{c|c} -c_B^T A_B^{-1} b & c_N^T - c_B^T A_B^{-1} A_N \\ \hline A_B^{-1} b & A_B^{-1} A_N \end{array}\right)$$

heißt verkürztes Simplex - Tableau zur Basis B.

Im Algorithmus 7.7 wird berechnet:

$$\left(\begin{array}{c|c} -c_0 & \bar{c}^T \\ \hline \bar{b} & \bar{A} \end{array}\right)$$

Beispiel:

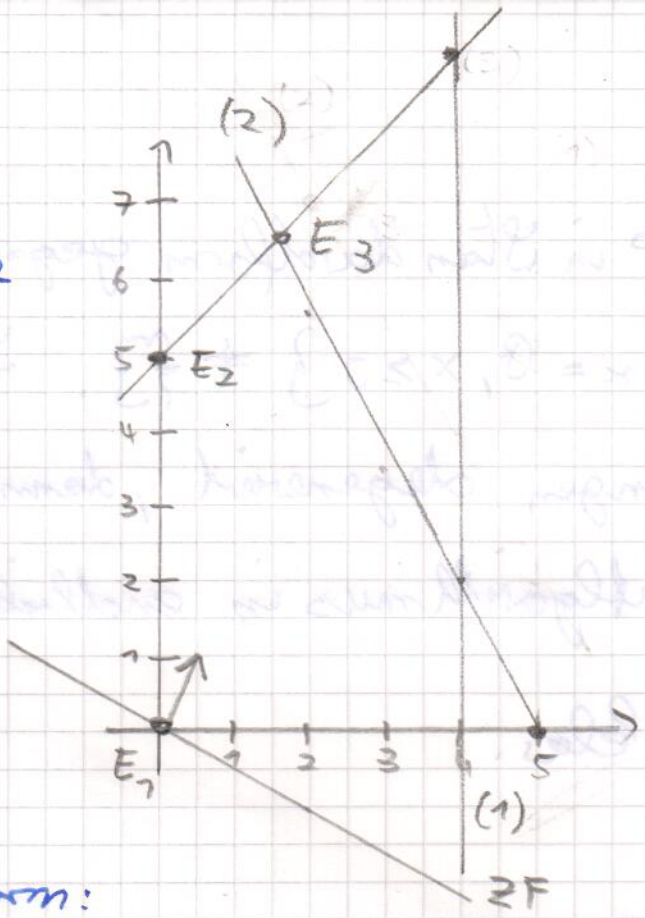
$\max x_1 + 2x_2$

$x_1 \leq 4$

$2x_1 + x_2 \leq 10$

$-x_1 + x_2 \leq 5$

$x_1, x_2 \geq 0$



In Standardform:

$\max (1,2,0,0,0)\ (x_1, \dots, x_5)^T$

$$\begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 2 & 1 & 0 & 1 & 0 \\ -1 & 1 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} = \begin{pmatrix} 4 \\ 10 \\ 5 \end{pmatrix} \qquad x_1, x_2 \dots, x_5 \geq 0$$

$x_3 = 4 - x_1 \qquad x_4 = 10 - 2x_1 - x_2 \qquad x_5 = 5 + x_1 - x_2$

$c_0 = \quad x_1 + 2x_2$

$E_1 = (0,0,4,10,5) \qquad E_2 = (0,5,4,5,0) \qquad E_3 = \left(\frac{5}{3}, \frac{20}{3}, \frac{7}{3}, 0, 0\right)$

	0	x_1	x_2
		1	2
x_3	4	1	0
x_4	10	2	1
x_5	5	-1	(1)

	-10	x_1	x_5
		3	-2
x_3	4	1	0
x_4	5	(3)	-1
x_2	5	-1	1

	-15	x_4	x_5
		-1	-1
x_3	$\frac{7}{3}$		
x_1	$\frac{5}{3}$		
x_2	$\frac{20}{3}$		

Optimallösung mit ZF-Wert 15

Satz 7.9:

Sei ein LP in Standardform gegeben mit $P = \{x : Ax = b, x \geq 0\} \neq \{\}$. Sind alle Basislösungen degeneriert, dann terminiert der Simplex-Algorithmus in endlich vielen Schritten.

Beweis: klar.

Die Phase I

Gegeben: LP in Standardform o.B.d.A $b \geq 0$

Hilfsproblem:

$$(LP_H) \quad \max \mathbb{1}^T Ax = \sum_{j=1}^{n} \left(\sum_{i=1}^{m} a_{ij} \right) x_j$$

$$Ax + Iy = b \qquad D\begin{pmatrix} x \\ y \end{pmatrix} = b$$

$$x, y \geq 0 \qquad x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}, \; y = \begin{pmatrix} y_{n+1} \\ \vdots \\ y_{n+m} \end{pmatrix}$$

$$\mathbb{1}^T Ax = \mathbb{1}^T (b - Iy) = \mathbb{1}^T b - \mathbb{1}^T y$$

Triviale **Startbasis**: $x = 0, \; y = b$

Sei $z = \begin{pmatrix} x \\ y \end{pmatrix}$ optimale Basislösung für (LP_H).

1.) Unzulässigkeit:

Falls $\pi^T A x < \pi^T b$, so gibt es keine Lösung mit $y = 0$, d.h. $P = \{\}$.

Sonst: $\pi^T A x = \pi^T b$ und $\pi^T A x = \pi^T b - \pi^T y$

$\Rightarrow \pi^T y = 0 \Rightarrow y = 0$ und x zulässig bzgl. P.

2) Trivialfälle:

Falls $B \cap \{n+1, \dots, n+m\} = \{\}$, dann ist $D_B = A_B$ und Rang $A_B = m \rightarrow$ Phase II

Falls $N \cap \{1, \dots, n\} = \{\}$, so gilt $m = n$ und $P = \{x\}$. STOP.

Andernfalls: $m < n \rightarrow$ Phase II

Die Basislösung $z^T = (z_B^T, z_N^T)$ mit $z_B = D_B^{-1} b$, $z_N = 0$ ist degeneriert:

$z_B^T = (y_{n_1}, \dots, y_{n_t}, x_{n_{t+1}}, \dots, x_{n_m})$

3) Entfernung künstlicher Variablen durch Pivotierung:

Falls es $r \in \{1,\dots,t\}$ und $s \in \{1,\dots,n\}$ gibt, so dass z_{qs} keine künstliche Variable ist und $\bar{a}_{rs} \neq 0$ gilt, dann wird eine Pivotoperation mit Element $\bar{a}_{rs}$ durchgeführt.

4.) Reduktion der Matrix:

Falls $\bar{a}_{rs} = 0$ für alle $r \in \{1,\dots,t\}$, $s \in \{1,\dots,n\}$ mit z_{qs} nicht-künstlich, dann hat

$z_B = D_B^{-1} b - D_B^{-1} D_N z_N$ folgende Form

$$\begin{pmatrix} y_{n_1} \\ \vdots \\ y_{n_t} \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix} - \left(0 \,\middle|\, *\right) \begin{pmatrix} x_N \\ y_N \end{pmatrix} \Leftrightarrow y_B = 0 - 0 x_N$$

$$\begin{pmatrix} x_{n_{t+1}} \\ \vdots \\ x_{n_m} \end{pmatrix} = \begin{pmatrix} * \\ \end{pmatrix} - \left(* \,\middle|\, *\right) \begin{pmatrix} x_N \\ y_N \end{pmatrix}$$

$\Rightarrow D_{1\cdot}, \dots, D_{t\cdot}$ können gestrichen werden

Mit $I = \{t+1,\dots,m\}$ und $J = \{j \in \{1,\dots,n\} : z_{q_j}$ ist nicht-künstlich$\}$ gilt:

$$x_B = (D_B^{-1} b)_I - \bar{D}_{IJ} x_N,$$

und $B = (j_{t+1}, \ldots, j_m)$ ist zulässige Basis von A_I.

Der revidierte Simplex-Algorithmus

In der Grundversion des Simplex-Algorithmus werden in einer Iteration i. allg. nicht alle Spalten von $\bar{A}$ benötigt. Der revidierte Simplex-Algorithmus berechnet nur die benötigten Spalten.

Gegeben sei ein LP in Standardform, Bezeichnungen wie bisher.

Algorithmus 7.10: Revidierter Simplex-Algorithmus

Gegeben seien A vollrangig, A_B^{-1}, $\bar{b}$, c und B.

(1) BTRAN (Backward Transformation)

Berechne $\pi^T := c_B^T A_B^{-1}$

(2) PRICE (Pivotspaltenauswahl)

Berechne die reduzierten Kostenkoeffizienten

$\bar{c}_j := (c_N^T)_j - \pi^T A_N e_j$ für $j = 1, \ldots, n-m$

und wähle einen Index s mit $\bar{c}_s > 0$.

(3) FTRAN (Forward Transformation)

Aktualisiere die Pivotspalte:

$\bar{a} := A_B^{-1} A_{\cdot q_s}$

(4) CHUZR (Pivotzeilen auswahl)

Berechne $\lambda_0 := \min\left\{\frac{\bar{b}_i}{\bar{a}_i}, \bar{a}_i > 0, i = 1, \dots, m\right\}$

und wähle einen Index $r \in \{1, \dots, m\}$ mit $\frac{\bar{b}_r}{\bar{a}_r} = \lambda_0$

(5) WRETA (Update der Basis)

Entferne das r-te Element von B und ersetze es durch q_s. Der neue Spaltenindexvektor heiße B'. Berechne $A_{B'}^{-1}$, $\bar{b} = A_{B'}^{-1} b$.

Abbruchkriterien:

zu (2): STOP, falls $\bar{c} \leq 0$ $\Rightarrow$ optimal

zu (4): STOP, falls $\bar{a} \leq 0$ $\Rightarrow$ unbeschränkt

Ansonsten: Nach (5) weiter mit (1) und $B := B'$.

Bemerkung 7.11: Vorteile der revidierten Simplex-Methode

1.) Wesentlich weniger Rechenaufwand, speziell bei Problemen mit erheblich mehr Variablen als Gleichungen.

2.) Die Matrix A kann auf externem Speicher oder (etwa bei dünn besetzten Matrizen) in speziellen Speicherstrukturen gehalten werden. Spalten von A werden je nach Bedarf generiert.

3.) A_B^{-1} muß nicht berechnet werden, sondern kann numerisch stabil gelöst werden.

4.) Man kann in jeder Iteration die Lösung für A_B^{-1} aufdatieren (Low-Rank-Updates). Dabei kann man nur die zum Update benötigten Informationen speichern und daraus jeweils A_B^{-1} aufbauen (Eta-File).

Spalten- und Zeilenauswahlregeln

Varianten von PRICE:

Sei $S := \{ j \in \{1, \dots, n-m\} : \bar{c}_j > 0 \}$

(1) Kleinster-Index-Regel:

Wähle $s := \min \{ j \in S \}$

(2) Kleinste-Variablenindex-Regel:

Wähle $s \in S$ so, dass $q_s = \min \{ q_j \in \mathbb{N} : j \in S \}$

(3) Steilster-Anstieg-Regel:

Wähle $s \in S$ so, dass $\bar{c}_s = \max \{ \bar{c}_j, j \in S \}$

(4) Größter-Fortschritt-Regel:

Berechne für jedes $j \in S$ zunächst

$$\lambda_0^j := \min \left\{ \frac{\bar{b}_i}{\bar{a}_{ij}}, \bar{a}_{ij} > 0, i = 1, \dots, m \right\}$$

und $g_j := \bar{c}_j \cdot \lambda_0^j$.

Wähle $s \in S$ so, dass $g_s = \max \{ g_j, j \in S \}$.

Variante von CHVÁTAL:

Definition 7.12:

Ein Vektor heißt lexikographisch positiv, falls seine erste von Null verschiedene Komponente positiv ist: $x \succ 0$.

$x \succ y \Leftrightarrow x - y \succ 0$.

Lexikographische Zeilenauswahlregel:

Wähle $r \in \left\{ i \in \{1, \dots, m\} : \frac{\bar{b}_i}{\bar{a}_{is}} = \lambda_0 \right\} =: R$ so, dass $\frac{1}{\bar{a}_{rs}} \left(A_B^{-1} A\right)_{r\cdot} = \operatorname{lexmin} \left\{ \frac{1}{\bar{a}_{is}} \left(A_B^{-1} A\right)_{i\cdot} : \bar{a}_{is} > 0,\ i \in R \right\}$

Satz 7.13:

Falls die lexikographische Zeilenauswahlregel angewendet wird, dann terminiert das Simplexverfahren in endlich vielen Schritten unabhängig von der Pivotspaltenauswahl.

Weitere Zeilenauswahlregeln:

2. Lexikographische Zeilenauswahlregel:

Wähle $r \in R$ so, dass $\frac{1}{\bar{a}_{rs}} (A_B^{-1})_{r\cdot} =$

$= \text{lexmin} \left\{ \frac{1}{\bar{a}_{is}} (A_B^{-1})_{i\cdot} : \bar{a}_{is} > 0, i \in \{1, \dots, m\} \right\}$

garantiert auch Endlichkeit des Simplex verfahrens.

Kleinste-Index-Regel:

Wähle $r := \min R$.

Kleinste-Variablenindex-Regel:

Wähle $r \in R$ so, dass $\eta_r = \min \{ \eta_i \in B, i \in R \}$.

Diese beiden Regeln allein können nicht Endlichkeit des Simplexverfahrens garantieren.

Aber Kleinste-Variablenindex als Spaltenauswahlregel zusammen mit Kleinste Variablenindex als Zeilenauswahlregel: Bland-Regel.

Laufzeitkomplexität des Simplexverfahrens

Zu allen diesen Auswahlregeln kennt man jeweils eine Klasse von Polyedern und Zielfunktionen, so dass der Simplexalgorithmus durch alle Ecken des Polyeders läuft.
Da die Anzahl der Ecken exponentiell mit der Anzahl der Variablen wächst, sagt man, dass das Simplexverfahren ein exponentielles Worst-Case-Verhalten hat.

Beispiel: Klee-Minty, 1972

$$\max \sum_{i=1}^{n} 2^{n-i} x_i$$

$$\sum_{j=1}^{i-1} 2^{i-j+1} x_j + x_i \leq 5^i \quad , i = 1, \dots, n$$

$$x \geq 0$$

Steilster Fortschritt: Eine Iteration
Steilster Anstieg: $2^n - 1$ Iterationen

In der Praxis: (average case)

Anzahl der Iterationen $\sim 3m$

$\sim \log n$

Bemerkung 7.13:

In der Phase II eines jeden Simplexverfahrens werden prinzipiell die folgenden Schritte durchgeführt:

0.) Starte mit zulässiger Ecke.

1.) Optimalitätsprüfung

2.) Bestimmung einer Anstiegsrichtung

3.) Prüfung auf Unbeschränktheit

4.) Bestimmung einer Schrittweite

5.) Übergang zu einer neuen zulässigen Ecke.

Weiter bei 1.)

8. Duales Simplexverfahren

Primales LP	Duales LP
(P) $\max c^T x$	(D) $\min u^T b$
$Ax = b$	$u^T A \geq c^T$
$x \geq 0$	

Sei $B \subset \{1, \dots, m\}$ mit A_B regulär eine Basis.

(P) $\max c^T x$	(D) $\min u^T b$
$x_B = A_B^{-1} b - A_B^{-1} A_N x_N$	$u^T A_B \geq c_B^T$
$x_B, x_N \geq 0$	$u^T A_N \geq c_N^T$

Basislösung:

$$x_B = A_B^{-1} b, \quad x_N = 0 \qquad u^T = c_B^T A_B^{-1}$$

primal zulässig, falls

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

dual zulässig, falls

$$u^T A_N = c_B^T A_B^{-1} A_N \geq c_N^T$$

$$\Leftrightarrow \bar{c}^T = c_N^T - c_B^T A_B^{-1} A_N \leq 0$$

Zielfunktionswert: $c^T x = c_B^T x_B = u^T A_B x_B = u^T b$

Satz 7.1:

Sei $P := \{u : u^T A \geq c^T\}$. Der Vektor u ist genau dann Ecke von P, falls u eine dual zulässige Basislösung ist.

Beweis:

Ecke $\Leftrightarrow$ Teilsystem mit reguläre Matrix ist mit Gleichheit erfüllt. ∎

Satz 8.2:

Ist A_B eine optimale Basis für (P) mit Optimallösung $x_B = A_B^{-1} b$, $x_N = 0$, Dann ist $u^T = c_B^T A_B^{-1}$ eine Optimallösung von (D).

A_B ist optimal genau dann, wenn A_B primal und dual zulässig ist.

Beweis:

Dualitätssatz. ∎

Die duale Basislösung $u^T = c_B^T A_B^{-1}$
mit A_B dual zulässig heißt Vektor
der Schattenpreise.

Interpretation:
Sei $x_B = A_B^{-1} b > 0$ optimal, nicht degeneriert.
$u^T = c_B^T A_B^{-1}$ ist Lösung von (D).

Betrachte eine Störung des Problems Δb,
so dass noch immer

$$A_B^{-1}(b + \Delta b) \geq 0$$

Dann ist

$$\hat{x} = \begin{pmatrix} A_B^{-1}(b+\Delta b) \\ 0 \end{pmatrix}$$ eine zulässige und
optimale Basis-Lösung des gestörten Problems

$$\begin{aligned} \max\ & c^T x \\ Ax &= b + \Delta b \qquad (P_\Delta) \\ x &\geq 0 \end{aligned}$$

denn $c^T \hat{x} = c_B^T A_B^{-1}(b+\Delta b) = u^T(b+\Delta b)$.

(optimal nach Dualitätssatz für gestörtes Problem)

Insbesondere gilt:

$$c^T \hat{x} = u^T(b + \Delta b) = u^T b + u^T \Delta b$$
$$= c^T x + u^T \Delta b$$

d.h. u^T zeigt an, wie stark sich der optimale Zielfunktionswert bei (kleinen zulässigen) Störungen von b ändert.

d.h. u^T beschreibt die Sensitivität des optimalen Zielfunktionswerts gegenüber (kleinen) Störungen von b.

Der duale Simplexalgorithmus

Prinzipiell kann man den Simplexalgorithmus aus Kapitel 7 (primale Simplexalgorithmus) auch auf das duale Problem anwenden, nachdem man dieses in Standardform transformiert hat.

Dies kann man aber auch als einen Algorithmus für das primale Problem auffassen mit den prinzipiellen Schritten aus Bemerkung 7.13, bei dem aber mit einer dual zulässigen Ecke gestartet wird, in 1.) die duale Optimalität, d.h. die primale Zulässigkeit überprüft wird und dann ein Abstieg der dualen Zielfunktion und ein Übergang zu einer neuen dual zulässigen Ecke durchgeführt wird. Der dualen Unbeschränktheit folgt primale Unzulässigkeit.

Algorithmus 8.3: Dualer-Simplex-Algorithmus

Input: $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$

Output: optimale Lösung des LP $\max c^T x$ mit $Ax = b$, $x \geq 0$

Phase I:

Bestimme eine dual zulässige Basis A_B von A.

Berechne $\bar{A} := A_B^{-1} A_N$, $\bar{b} := A_B^{-1} b$, $\bar{c}^T := c_N^T - c_B^T A \leq 0$

Phase II:

(II.1) **Optimalitätsprüfung**

Gilt $\bar{b}_i \geq 0$, $i = 1, \ldots, m$, so ist die gegenwärtige Basislösung optimal.

Setze $x_B := \bar{b}$, $x_N := 0$. STOP.

(II.2) **Bestimmung der Pivotzeile**

Wähle einen Index r mit $\bar{b}_r < 0$.

(II.3) **Prüfung auf Unbeschränktheit**

Gilt $\bar{A}_{r\cdot} \geq 0$, so ist das duale Problem unbeschränkt.

(Dann sei $\gamma > 0$, dann ist $y_\gamma^T := y^T + \gamma e_r^T A_B^{-1}$ dual zulässig:

$$y_\gamma^T A = \begin{pmatrix} y^T A_B + \gamma e_r^T \\ y^T A_N + \gamma \underbrace{\bar{A}_{r\cdot}}_{\geq 0} \end{pmatrix}^T \geq \begin{pmatrix} c_B^T \\ c_N^T \end{pmatrix} = c^T$$

und

$$y_\gamma^T b = y^T b + \gamma e_r^T A_B^{-1} b = y^T b + \gamma \underbrace{\bar{b}_r}_{<0}$$ beliebig klein)

Dann ist das LP $\max\{c^T x : Ax = b, x \geq 0\}$ unzulässig. STOP.

(II.4) Bestimmung der Schrittweite

Berechne $\lambda_0 := \min\left\{\frac{\bar{c}_j}{\bar{a}_{rj}}, \bar{a}_{rj} < 0, j = 1, \dots, n-m\right\} \geq 0$

(II.5) Bestimmung der Pivotspalte

Wähle einen Index $s \in \{1, \dots, n-m\}$ mit $\frac{\bar{c}_s}{\bar{a}_{rs}} = \lambda_0$

(II.6) Pivotoperation

Bilde B' durch Entfernen von p_r und Hinzufügen von q_s.

Berechne $A_{B'}^{-1} = E A_B^{-1}$ mit E wie in Satz 7.4,

berechne $\bar{A}, \bar{b}, \bar{c}$ neu,

gehe zu (II.1).

Bemerkung 8.4:

- Primales Simplexalgorithmus, Phase II:
  Starte primal zulässig, iteriere so lange,
  bis auch dual zulässig.
- Duales Simplexalgorithmus, Phase II:
  Starte dual zulässig, iteriere so lange,
  bis auch primal zulässig.

Wann nimmt man die primale und wann die duale Variante?

Ziel: Vermeidung der Phase I

- Wenn man eine primal zulässige Lösung kennt,
  nimmt man den primalen Simplexalgorithmus.
- Wenn man eine dual zulässige Lösung kennt,
  nimmt man den dualen Simplexalgorithmus.

Diese Fälle treten bei Änderungen am Problem mit anschließender Re-Optimierung auf, siehe Kapitel 9.

9. Re-Optimierung

Situation:

Wir haben ein lineares Optimierungsproblem gelöst. Im Nachhinein wird das Problem verändert, z.B. durch Änderung der Zielfunktion oder zusätzliche Variablen oder Beschränkungen.
Wie ändert sich dann die Optimallösung?

Wir betrachten ein LP in Standardform

$$\max c^T x$$
$$Ax = b$$
$$x \geq 0$$

Sei A_B optimale Basis.

9.1 Änderung der rechten Seite

Neue rechte Seite: $b' = b + \Delta b$

Neue Basislösung zur Basis A_B:

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

1. Fall: $x_B' \geq 0$.

Dann ist (x_B', x_N') mit $x_N' = 0$ optimal, da sich die reduzierten Kosten nicht geändert haben: $\bar{c}^T = c_N^T - c_B^T A_B^{-1} A_N \leq 0$.

2. Fall: $x_B' \not\geq 0$.

Die neue Basislösung ist primal unzulässig. Die Optimalitätsbedingung $\bar{c} \leq 0$, d. h. die duale Zulässigkeit, ist aber weiterhin erfüllt.

$\Rightarrow A_B$ ist eine dual zulässige Basis,

wir können direkt mit Phase II des dualen Simplexalgorithmus zur Neuberechnung einer Optimallösung beginnen.

9.2 Änderung der Zielfunktion

Neue Zielfunktion: $c' = c + \Delta c$

A_B optimale Basis, $x_B = A_B^{-1} b - A_B^{-1} A_N x_N$

Neue Kosten der Basislösung

$$(c+\Delta c)^T x = (c_B + \Delta c_B)^T A_B^{-1} b + \left((c_N + \Delta c_N)^T - (c_B + \Delta c_B)^T A_B^{-1} A_N\right) x_N$$

$$= c_B^T \bar{b} + \Delta c_B^T \bar{b} + \bar{c}^T x_N + \underbrace{\left(\Delta c_N^T - \Delta c_B^T A_B^{-1} A_N\right)}_{=: \Delta \bar{c}^T} x_N$$

Basislösung: $x_N = 0$, $x_B = A_B^{-1} b$ bleibt primal zulässig.

Falls $\Delta c_B = 0$:

Der Zielfunktionswert bleibt unverändert.

1. Fall: Neue reduzierte Kosten $\bar{c} + \Delta \bar{c} \leq 0$

$\Rightarrow$ duale Zulässigkeit $\Rightarrow$ Optimalität.

2. Fall: Falls j existiert mit $\bar{c}_j + \Delta \bar{c}_j > 0$:

$\Rightarrow$ duale Zulässigkeit verletzt. Wende Phase II des primalen Simplexalgorithmus an mit primal

zulässige Anfangs-Basis A_B.

Beispiel: (vgl. Beispiel in Kapitel 7)

$\max x_1 + 2x_2$

$x_1 + x_3 = 4$

$2x_1 + x_2 + x_4 = 10$

$-x_1 + x_2 + x_5 = 5$

$x_1, x_2, x_3, x_4, x_5 \geq 0$

optimale Basislösung:

$\left(\frac{5}{3}, \frac{20}{3}, \frac{7}{3}, 0, 0\right)$

		x_4	x_5
	-15	-1	-1
x_3	$\frac{7}{3}$	$-\frac{1}{3}$	$\frac{1}{3}$
x_1	$\frac{5}{3}$	$\frac{1}{3}$	$-\frac{1}{3}$
x_2	$\frac{20}{3}$	$\frac{1}{3}$	$\frac{2}{3}$

$c' = c + \Delta c$ $\quad \Delta c = \begin{pmatrix} \delta \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$ $\quad \Delta c_B = \begin{pmatrix} 0 \\ \delta \\ 0 \end{pmatrix}$ $\quad \Delta c_N = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$

$c_1' = c_1 + \delta$

$\left(\bar{c} + \Delta\bar{c}\right)^T = (-1, -1) - \delta\left(\frac{1}{3}, -\frac{1}{3}\right) = \left(-1 - \frac{1}{3}\delta, -1 + \frac{1}{3}\delta\right) \leq 0$

$\Leftrightarrow -3 \leq \delta \leq 3$

Für $-3 \leq \delta \leq 3$ bleibt die Lösung optimal.

9.3 Änderung einer Spalte von A

Neuer Spaltenvektor: $A'_{\cdot j} = A_{\cdot j} + \Delta A_{\cdot j}$

1. Fall: $A_{\cdot j}$ Basisspalte

→ Berechne neue Basis-Inverse

2. Fall: $A_{\cdot j}$ Nichtbasisspalte

→ primale Zulässigkeit bleibt erhalten

$$x'_B = A_B^{-1} b - A_B^{-1} A'_N x'_N$$
$$= A_B^{-1} b \geq 0 \qquad x'_N = 0$$

Reduzierte Kosten dann: mit $u^T A_B = c_B^T$

$$\bar{c}'_j = c_j - c_B^T A_B^{-1} (A_{\cdot j} + \Delta A_{\cdot j})$$

$$= \bar{c}_j - c_B^T A_B^{-1} \Delta A_{\cdot j} = \bar{c}_j - u^T \Delta A_{\cdot j}$$

Falls $\bar{c}'_j \leq 0$ → Optimalität

Falls $\bar{c}'_j > 0$ → Weiter mit Phase II des primalen Simplexalgorithmus

Beispiel: max $-x_1 - 2x_2$

$$-x_1 - x_2 + x_3 = -2$$
$$-2x_1 - x_2 + x_4 = -3$$
$$x_1, x_2, x_3, x_4 \geq 0$$

Basis (x_1, x_4) Nichtbasis (x_2, x_3) optimal

$A_B = \begin{pmatrix} -1 & 0 \\ -2 & 1 \end{pmatrix} = A_B^{-1}$ $x_B = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$ $\bar{c}^T = (-1 \; -1) \leq 0$

Ändere Spalte 2: $A'_{\cdot 2} = \begin{pmatrix} -1 + \alpha \\ -1 + \beta \end{pmatrix}$

$\bar{c}_2 = -1$ $\bar{c}'_2 = -1 - (-1 \; 0)\begin{pmatrix} -1 & 0 \\ -2 & 1 \end{pmatrix}\begin{pmatrix} \alpha \\ \beta \end{pmatrix} = -1 - \alpha$

d.h. für $\alpha \geq -1$ β bel. bleibt Optimalität erhalten

3.4 Hinzufügen von Variablen

Neue Variable x_{n+1}, Spalte $A_{\cdot n+1}$, Zf.-Koeff c_{n+1}

$\bar{c}_{n+1} = c_{n+1} - c_B^T A_B^{-1} A_{\cdot n+1}$

$\leq 0 \rightarrow$ Optimalität

$> 0 \rightarrow$ dual unzulässig, primal weiter zulässig

$\rightarrow$ primale Phase II

3.5 Hinzufügen von Restriktionen

Neue Gleichung: $A_{m+1} \cdot x = b_{m+1}$

Annahme: aktuelle Lösung verletzt neue Gleichung

Schlupfvariable $x_{n+1} \geq 0$ mit

$A_{m+1,\cdot} x + x_{n+1} = b_{m+1}$, falls $A_{m+1,\cdot} \bar{x} > b_{m+1}$

$A_{m+1,\cdot} x - x_{n+1} = b_{m+1}$, falls $A_{m+1,\cdot} \bar{x} < b_{m+1}$

x_{n+1} wird Basisvariable mit Wert

$$\bar{b}_{n+1} = -| b_{m+1} - A_{m+1,\cdot} \bar{x} |$$

Neue Basis: $\begin{pmatrix} A_B & 0 \\ A_{m+1,B} & 1 \end{pmatrix}$

Neue Basisinverse $\begin{pmatrix} A_B^{-1} & 0 \\ -A_{m+1,B} A_B^{-1} & 1 \end{pmatrix}$

$\bar{b}_{m+1} < 0 \rightarrow$ neue Basis primal nicht zulässig

$$\bar{c}^{T} = c_N^T - c_B^T A_B^{-1} A_N$$

$$= c_N^T - \left(c_B^T A_B^{-1} \quad 0 \right) \begin{pmatrix} A_N \\ A_{m+1,N} \end{pmatrix}$$

$$= c_N^T - c_B^T A_B^{-1} A_N = \bar{c}^T \leq 0$$

$\rightarrow$ neue Basis weiterhin dual zulässig

$\rightarrow$ Phase II des dualen Simplexalgorithmus

9.6 Hinzufügen einer Ungleichung

Gegeben: $\max c^T x$ mit optimaler Basis B

$Ax = b$

$x \geq 0$

und Tableau

$$\left(\begin{array}{c|c} -c_B^T A_B^{-1} b & c_N^T - c_B^T A_B^{-1} A_N \\ \hline A_B^{-1} b & A_B^{-1} A_N \end{array}\right)$$

Modifiziertes Problem: $\max c^T x$

Neue Ungleichung:

$a^T x \leq \beta$ $\quad\Rightarrow\quad$ $a^T x + x_{n+1} = \beta$

$Ax = b$

$x \geq 0$

Neue Variable: x_{n+1} wird Basisvariable, $B' = B \cup \{n+1\}$

$$A_{B'} = \begin{pmatrix} A_B & 0 \\ a_B^T & 1 \end{pmatrix} \qquad A_{B'}^{-1} = \begin{pmatrix} A_B^{-1} & 0 \\ -a_B^T A_B^{-1} & 1 \end{pmatrix} \qquad A_{N'} = \begin{pmatrix} A_N \\ a_N^T \end{pmatrix}$$

$$c_{B'}^T = (c_B^T \; 0)$$

Neue Dualvariable:

$$u'^T = c_{B'}^T A_{B'}^{-1} = (c_B^T A_B^{-1} \; 0)$$

$$b' = \begin{pmatrix} b \\ \beta \end{pmatrix}$$

$$N' = N$$

Neue reduzierte Kosten:

$$\bar{c}'^T = c_{N'}^T - c_{B'}^T A_{B'}^{-1} A_{N'} = c_N^T - c_B^T A_B^{-1} A_N = \bar{c}^T$$

Bleiben unverändert ≤ 0 $\Rightarrow$ Bleibt dual zulässig

Neues Tableau:

$-c_B^T A_B^{-1} b$	$c_N^T - c_B^T A_B^{-1} A_N$
$A_B^{-1} b$	$A_B^{-1} A_N$
$-a_B^T A_B^{-1} b + \beta$	$-a_B^T A_B^{-1} A_N + a_N^T$

Primale Zulässigkeit: • bleibt $x_B = A_B^{-1} b \geq 0$

• Falls $x_{n+1} = -a_B^T A_B^{-1} b + \beta \geq 0$ $\Rightarrow$ primal zulässig, x_B optimal
(d.h. neue Ungleichung ist erfüllt)

• Falls $x_{n+1} = -a_B^T A_B^{-1} b + \beta < 0$ $\Rightarrow$ primal unzulässig
(d.h. neue Ungleichung ist verletzt) weiter mit <u>dualer Phase II</u>

10. Innere - Punkte - Methoden

10.1 Polynomiale Algorithmen zu linearen Optimierung

Simplex - Algorithmus:

Im Worst-Case: 2^n Iterationen für LP mit n Variablen und 2n Ungleichungen: schlechtes Verhalten

Im Mittel: gutes Verhalten

Ellipsoid - Methode (Khachiyan 1979)

Prinzipieller Ablauf:

Input: Polytop P

Output: Punkt in P oder Nachweis, dass $P = \{\}$

(1) Konstruiere ein Ellipsoid E_0, das P enthält. $k := 0$

(2) Ist das gegenwärtige Ellipsoid E_k zu klein,

STOP: $P = \{\}$.

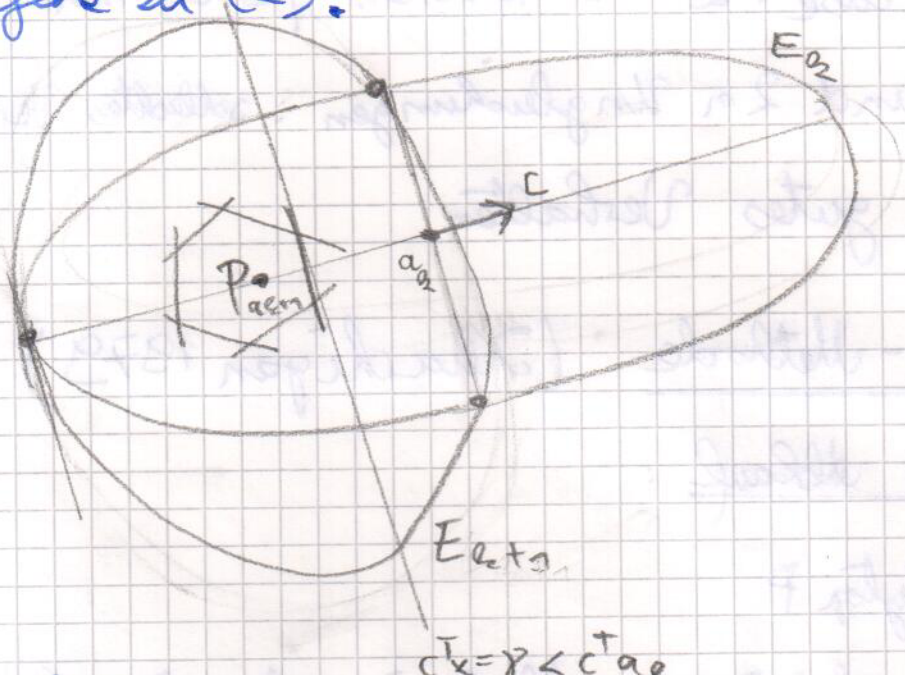
(3) Teste, ob der Mittelpunkt a_k von E_k in P enthalten ist.

(4) Gilt $a_k \in P$: STOP.

(5) Gilt $a_k \notin P$, dann gibt es eine P definierende Ungleichung $c^T x \leq \gamma$, die von a_k verletzt wird:

$c^T a_k > \gamma$. Sei $E_k' := E_k \cap \{x : c^T x \leq c^T a_k\}$ das Halbellipsoid von E_k, das P enthält.

Konstruiere das Ellipsoid kleinsten Volumens, das E_k' enthält, nenne es E_{k+1}, setze $k := k+1$, gehe zu (2).



$$\frac{\text{vol } E_{k+1}}{\text{vol } E_k} \leq e^{-\frac{1}{2n}} < 1$$

Falls P ein positives Volumen hat, wird dieses irgendwann unterschritten. Nach endlich vielen Schritten ist der Mittelpunkt von E_k in P oder $P = \{\}$.

Die Ellipsoid findet entweder einen Punkt in P oder beweist, dass P leer ist.

Die maximale Iterationszahl der Ellipsoid-Methode ist polynomial in m, n. Alle Iterationen werden in polynomialer Zeit ausgeführt.

Die Ellipsoid-Methode weist also in polynomial vielen Iterationen nach, dass P leer ist oder liefert einen Punkt in P. Daraus lässt sich (durch binäre Suche) ein Algorithmus formulieren, der lineare Programme in polynomialer Zeit löst.

Damit gezeigt: lineare Programme können in polynomialer Zeit gelöst werden.

In der Praxis ist diese Methode dem Simplex-Algorithmus weit unterlegen.

1984, Karmakar: theoretisch polynomiales Verfahren, auch effizientes Verhalten in der Praxis
Weiterentwicklung der Idee → Innere-Punkte-Verfahren

10.2 Innere-Punkte-Methoden

Betrachte das LP

$\min c^T x$ $\quad$ ab jetzt: min

$Ax = b$ $\quad$ $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$

$x \geq 0$ $\quad$ $c \in \mathbb{R}^n$

Wir wissen: Für optimales $x \in \mathbb{R}^n$ ist notwendig und hinreichend, dass es $y \in \mathbb{R}^m$ gibt mit

(1) $Ax = b$, $x \geq 0$ $\quad$ primale Zulässigkeit

(2) $A^T y + z = c$, $z \geq 0$ $\quad$ duale Zulässigkeit

(3) $z^T x = 0$ $\quad$ Komplementarität

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

(1) und (2) sind linear, (3) ist nicht linear.

$F := \{ x, y, z : Ax = b, x \geq 0, A^T y + z = c, z \geq 0 \}$

(durch die d linearen Bedingungen beschriebenes Polyeder)

$F_0 := \{ x, y, z : Ax = b, A^T y + z = c, x > 0, z > 0 \}$

(innere Punkte von F)

Die Bedingungen (1)-(3) sind ein Spezialfall der KKT-Bedingungen: Optimalitätsbedingungen für nicht lineare Optimierungsprobleme.

Satz 10.1: KKT-Bedingungen

Sei $\min f(x) : g(x) = 0,\ h(x) \geq 0$
mit $f: \mathbb{R}^n \to \mathbb{R}$, $g: \mathbb{R}^n \to \mathbb{R}^m$, $h: \mathbb{R}^n \to \mathbb{R}^k$,
f, g, h stetig differenzierbar, ein nichtlineares Optimierungsproblem. Sei $x^* \in \mathbb{R}^n$ ein lokales Minimum, dann existieren „Lagrange-Multiplikatoren" $y \in \mathbb{R}^m$, $z \in \mathbb{R}^k$, die eine Lagrangefunktion $\mathcal{L}(x, y, z) = f(x) - y^T g(x) - z^T h(x)$
definieren. Es gelten die notwendigen Bedingungen:

(1) $g(x^*) = 0,\ h(x^*) \geq 0$.

(2) $\frac{d}{dx}\mathcal{L}(x^*, y, z) = 0$, d.h. $\left.\frac{\partial f}{\partial x} - y^T \frac{\partial g}{\partial x} - z^T \frac{\partial h}{\partial x}\right|_{x=x^*} = 0$,
$z \geq 0$,

(3) $z^T h(x^*) = 0$.

Korollar 10.2:

Die KKT-Bedingungen angewendet auf $\min c^T x : Ax = b, x \geq 0$ ergeben die Bedingungen (1)-(3).

Beweis:

$f(x) = c^T x$, $g(x) = Ax - b$, $h(x) = x$

(1) $Ax = b$, $x \geq 0$

(2) $\mathcal{L}(x, y, z) = c^T x - y^T(Ax - b) - z^T x$

$\frac{\partial}{\partial x} \mathcal{L}(x, y, z) = c^T - y^T A - z^T = 0$

$\Rightarrow A^T y + z = c$, $z \geq 0$

(3) $z^T x = 0$. ∎

Satz 10.3:

Sei $f \in C^2(\mathbb{R}^n, \mathbb{R})$ und strikt konvex. Dann sind die KKT-Bedingungen auch hinreichend und eine Lösung ist eindeutig.

Sat hier erfüllt: (1)-(3) sind hinreichend und notwendig für Optimalität.

Lösung von nichtlinearen Gleichungen mit dem Newton-Verfahren

$$F(w) = 0 \,, \quad F: \mathbb{R}^N \to \mathbb{R}^N \,, \quad F \in C^2$$

Algorithmus 10.4:

Startwert: w^0

Iteration: $w^{k+1} = w^k + \Delta w^k$, $k = 0, 1, 2, \ldots$

Das **Inkrement** Δw^k berechnet man aus der Linearisierung von F an der Stelle w^k:

$$F(w) \doteq F(w^k) + J(w^k)(w - w^k)$$

$J = F'$: Jacobi-Matrix

Man löst das **linearisierte Problem**

$$F(w^k) + J(w^k)\,\Delta w^k = 0$$

(d.h. das lineare Gleichungssystem durch Zerlegung von $J(w^k)$, formal: $\Delta w^k = -J(w^k)^{-1} F(w^k)$)

Das Newton-Verfahren konvergiert lokal quadratisch.

Durch Verkürzung der Schritte $w^{k+1} = w^k + \alpha^k \Delta w^k$ mit $\alpha^k \in (0, 1]$ kann man globale Konvergenz erreichen.

Bemerkung 10.5:

Lineare Probleme $F(x) = Ax - b = 0$ werden vom Newton-Verfahren in einer Iteration gelöst.
Lineare Gleichungen in einem nicht linearen Problem werden in einer Iteration erfüllt und bleiben in allen weiteren Iterationen erfüllt.

Notation:

Vektor $e = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}$ Matrix $X = \begin{pmatrix} x_1 & & 0 \\ & \ddots & \\ 0 & & x_n \end{pmatrix}$ für einen Vektor $x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$

Anwendung:

$$e^T x = \sum_{i=1}^{n} x_i$$

$$Xe = x$$

$$X^{-1} = \begin{pmatrix} \frac{1}{x_1} & & 0 \\ & \ddots & \\ 0 & & \frac{1}{x_n} \end{pmatrix}$$

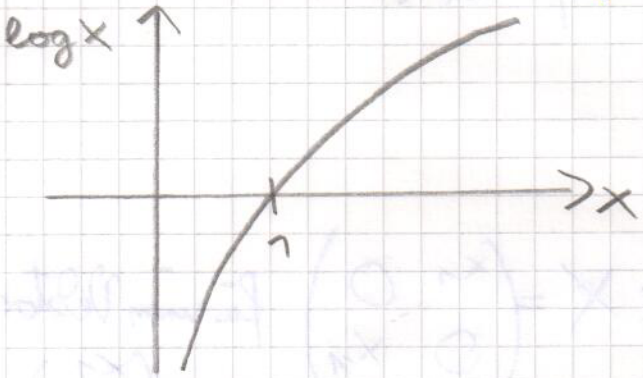
$$XZ = \begin{pmatrix} x_1 z_1 & & 0 \\ & \ddots & \\ 0 & & x_n z_n \end{pmatrix}$$

$$e^T X Z e = \sum_{i=1}^{n} x_i z_i = x^T z$$

Barriere-Ansatz

Wir wollen die Verletzung von Ungleichungsnebenbedingungen durch einen Straf-/Penalty-Term in der Zielfunktion verhindern.

Brauchbare Straffunktion: log-Barriere



$\log x \to -\infty$ für $x \to 0$

d.h. $-\tau \log x$ ($\tau > 0$ konstant)

ist eine Barriere bei $x = 0$

Beispiel:

$\min x : x \geq 0 \quad , x \in \mathbb{R}$ Lösung: $x = 0$

Barriere-Ansatz: $\min x - \tau \log x$

Lösung: Ableitung der Zielfunktion nullsetzen

$1 - \tau \frac{1}{x} = 0 \Rightarrow x = \tau$

$\tau \to 0 \Rightarrow x \to 0$

Anwendung des Barrieré-Ansatzes auf LP

Annahme: Es gibt innere Punkte, d.h.

$$P = \{x : Ax = b, x > 0\} \neq \{\}$$

bzw.

$$\bar{P} = \{y, z : A^T y + z = c, z > 0\} \neq \{\}$$

Primales Barrieré-Problem

Sei $\tau > 0$

$$\min c^T x - \tau \sum_{i=1}^{n} \log x_i \qquad (P_\tau)$$
$$Ax = b$$

Satz 10.6:

x ist Lösung von (P_τ) genau dann, wenn es $y \in \mathbb{R}^m$ und $z \in \mathbb{R}^n$ gibt, so dass

$$Ax = b$$
$$A^T y + z = c \qquad (KKT\text{-}P_\tau)$$
$$z = \tau X^{-1} e \quad \left(z_i = \frac{\tau}{x_i}\right)$$

$(KKT\text{-}P_\tau)$ ist ein nicht lineares Gleichungssystem für (x, y, z).

Beweis:

KKT-Bedingungen für (P_τ):

(1) $Ax = b$

(2) $\mathcal{L}(x,y) = c^T x - \tau \sum_{i=1}^{n} \log x_i - y^T(Ax-b)$

$\frac{\partial}{\partial x_i}\mathcal{L}(x,y) = c_i - \tau \frac{1}{x_i} - (y^T A)_i = 0$

d.h. $c - \tau X^{-1} e - y^T A = 0$

Definiere $z := \tau X^{-1} e \in \mathbb{R}^n$

Dann ist $A^T y + z = c$. ∎

Duales Barriere-Problem

Sei $\tau > 0$.

$$\max y^T b + \tau \sum_{i=1}^{n} \log z_i \qquad (D_\tau)$$
$$A^T y + z = c$$

Satz 10.7:

(y,z) ist Lösung von (D_τ) genau dann, wenn $x \in \mathbb{R}^n$ existiert mit (KKT-D_τ)

$A^T y + z = c$, $Ax = b$, $x = \tau Z^{-1} e$ $\left(x_i = \tau \frac{1}{z_i}\right)$

(KKT- D_τ) ist ein nichtlineares Gleichungssystem für (x, y, z).

Beweis:

KKT- Bedingungen für (D_τ):

(1) $A^T y + z = c$

(2) $\mathcal{L}(y, z, x) = b^T y + \tau \sum_{i=1}^{n} \log z_i - x^T (A^T y + z - c)$

$\frac{\partial \mathcal{L}}{\partial y} = b^T - x^T A^T = 0 \Rightarrow Ax = b$

$\frac{\partial \mathcal{L}}{\partial z_i} = \tau \cdot \frac{1}{z_i} - x_i = 0 \Rightarrow x = \tau Z^{-1} e$ ∎

Korollar 10.8:

Die KKT- Bedingungen für (P_τ) und (D_τ) sind identisch:

$x = \tau Z^{-1} e \Leftrightarrow z = \tau X^{-1} e$

Außerdem äquivalent: $\Leftrightarrow XZe = \tau e$

d.h. $x_i z_i = \tau$

Dazu gehört das folgende System:

Definition 10.9:

Das System

$$A^T y + z - c = 0$$
$$Ax - b = 0 \qquad (PD_\tau)$$
$$XZe - \tau e = 0$$

heißt das primal-duale System in den Variablen (x, y, z).

Bemerkung: ohne Ungleichungen

(PD_0) ist das eigentliche KKT-System des LP.

(PD_τ) heißt auch gestörtes KKT-System.

Ansatz: Newton-Verfahren für (PD_τ), $\tau > 0$

Startpunkt in Innen von F: $(x^0, y^0, z^0) \in F_0$,

d.h. $Ax_0 = b$, $A^T y_0 + z_0 = c$, $x_0, z_0 > 0$.

Die linearen Bedingungen bleiben während des ganzen Newton-Verfahrens erfüllt.

Linearisierung von (PD_τ):

$$\begin{pmatrix} Ax - b \\ A^T y + z - c \\ XZe - \tau e \end{pmatrix} + \underbrace{\begin{pmatrix} A & 0 & 0 \\ 0 & A^T & I \\ Z & 0 & X \end{pmatrix}}_{=: J} \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta z \end{pmatrix}$$

J ist regulär, wenn $\operatorname{Rang} A = m$.

Newtonschritt $\begin{pmatrix} x^{k+1} \\ y^{k+1} \\ z^{k+1} \end{pmatrix} = \begin{pmatrix} x^k \\ y^k \\ z^k \end{pmatrix} + \alpha^k \begin{pmatrix} \Delta x^k \\ \Delta y^k \\ \Delta z^k \end{pmatrix}$

mit Schrittweite α^k

Inkrement $(\Delta x^k, \Delta y^k, \Delta z^k)$

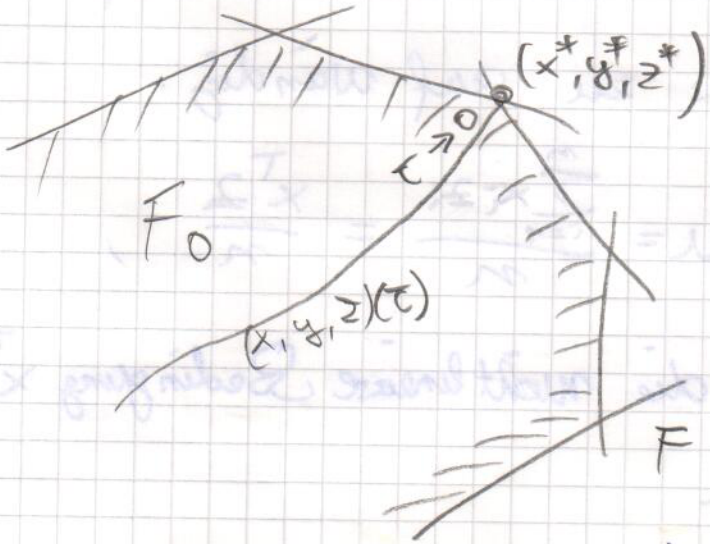
$$\begin{pmatrix} 0 \\ 0 \\ x^k z^k e - \tau e \end{pmatrix} + J \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta z \end{pmatrix} = 0$$

Wenn das Newton-Verfahren konvergiert, dann gegen einen Punkt auf dem „zentralen Pfad":

Definition 10.10:

Der **zentrale Pfad** ist die von τ abhängige Kurve der Lösungen von (KKT-PD$_\tau$), $\tau > 0$:

$\{ x(\tau), y(\tau), z(\tau)$, die (KKT-PD$_\tau$) erfüllen $\}$.



Auf dem zentralen Pfad gilt:

$x_i z_i = \tau$, $i = 1, \dots, n$

d.h. alle $x_i z_i$ sind gleich

Mögliches Verfahren: Fortsetzungsverfahren, um Konvergenz des Newton-Verfahrens zu erreichen

- Starte mit $(x^0, y^0, z^0) \in F_0$, $\tau_0 > 0$
- Löse (PD_τ) für eine Folge $\{\tau_k\}$, $\tau_0 > \tau_1 > \ldots > 0$ mit dem Newton-Verfahren, wobei von der Lösung des letzten Problems gestartet wird.
- Für $\tau_k \to 0$ geht $(x(\tau_k), y(\tau_k), z(\tau_k)) \to (x^*, y^*, z^*)$.

Warum nicht einfach Newton-Verfahren für $\tau = 0$?
Weil dann die Strafterme, d.h. die Bedingungen $x \geq 0$ und $z \geq 0$ nicht berücksichtigt werden, kann auch gegen unzulässige Punkte konvergieren.

Wir wollen für $\tau > 0$ das Newton-Verfahren nicht austerieren: zu aufwändig.

Dualitätsmaß: $\mu = \frac{\sum_{i=1}^{n} x_i z_i}{n} = \frac{x^T z}{n}$,

beschreibt, wie stark die nichtlineare Bedingung $x^T z = 0$ noch verletzt ist.

Wähle $\tau = \sigma \cdot \mu$, $\sigma \in [0; 1]$

Schritte werden bestimmt durch Lösen von

$$\begin{pmatrix} 0 \\ 0 \\ xz_\varepsilon + \sigma\mu \end{pmatrix} + J \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta z \end{pmatrix} = 0$$

- Für $\sigma = 1$ bekommen wir Zentrierungsschritte, also Newton-Schritte hin zum Punkt $(x(\mu), y(\mu), z(\mu))$ auf dem zentralen Pfad: $x(\mu)_i \cdot z(\mu)_i := \mu$.
  Das Dualitätsmaß μ wird dabei nicht reduziert.
- Für $\sigma = 0$ bekommen wir Newton-Schritte für $\tau = 0$, also ohne Straftern. Diese heißen Tangentialschritte, sie sind tangential zum zentralen Pfad.
  Sie reduzieren das Dualitätsmaß.

Schrittweite $\alpha \in (0;1]$

Damit die Iterierten im Innern F_0 bleiben, müssen wir die Schritte gef. verkürzen:

$$\begin{pmatrix} x^{k+1} \\ y^{k+1} \\ z^{k+1} \end{pmatrix} = \begin{pmatrix} x^k \\ y^k \\ z^k \end{pmatrix} + \alpha \begin{pmatrix} \Delta x^k \\ \Delta y^k \\ \Delta z^k \end{pmatrix}$$

$\alpha \in (0;1]$ so, dass $\begin{pmatrix} x^{k+1} \\ y^{k+1} \\ z^{k+1} \end{pmatrix} \in F_0$, d.h. $x^{k+1} > 0$, $z^{k+1} > 0$

$x_i + \alpha \Delta x_i > 0 \Rightarrow \alpha < \frac{-x_i}{\Delta x_i}$ für $\Delta x_i < 0$

($\alpha > \frac{-x_i}{\Delta x_i} < 0$ für $\Delta x_i > 0$ immer erfüllt)

$z_i + \alpha \Delta z_i > 0 \Rightarrow \alpha < \frac{-z_i}{\Delta z_i}$ für $\Delta z_i < 0$

($\alpha > \frac{-z_i}{\Delta z_i} < 0$ für $\Delta z_i > 0$ immer erfüllt)

Also:

$$\alpha < \min \left\{ \frac{-x_i}{\Delta x_i}, \Delta x_i < 0, \frac{-z_i}{\Delta z_i}, \Delta z_i < 0 \right\} = \alpha^*$$

und $\alpha \leq 1$

also $\alpha = \min \{1; 0{,}9\alpha^*\}$.

Wann macht man Zentrierungs- und wann Tangentialschritte? Wird bestimmt durch die folgenden <u>Nachbarschaften</u>:

- **Nähe zum Zentralen Pfad**

$$N_2(\theta) = \left\{ \begin{pmatrix} x \\ y \\ z \end{pmatrix} : \| XZe - \mu e \|_2 \leq \theta \cdot \mu \right\}$$

$0 < \theta < 1$

Die Bedingung $\begin{pmatrix} x \\ y \\ z \end{pmatrix} \in N_2(\theta)$ ist sehr restriktiv

für große n: $\sum_{i=1}^{n} (x_i z_i - \mu)^2 \leq \theta^2 \mu^2$

bzw. $\sum_{i=1}^{n} \left(\frac{x_i z_i}{\mu} - 1 \right)^2 \leq \theta^2$

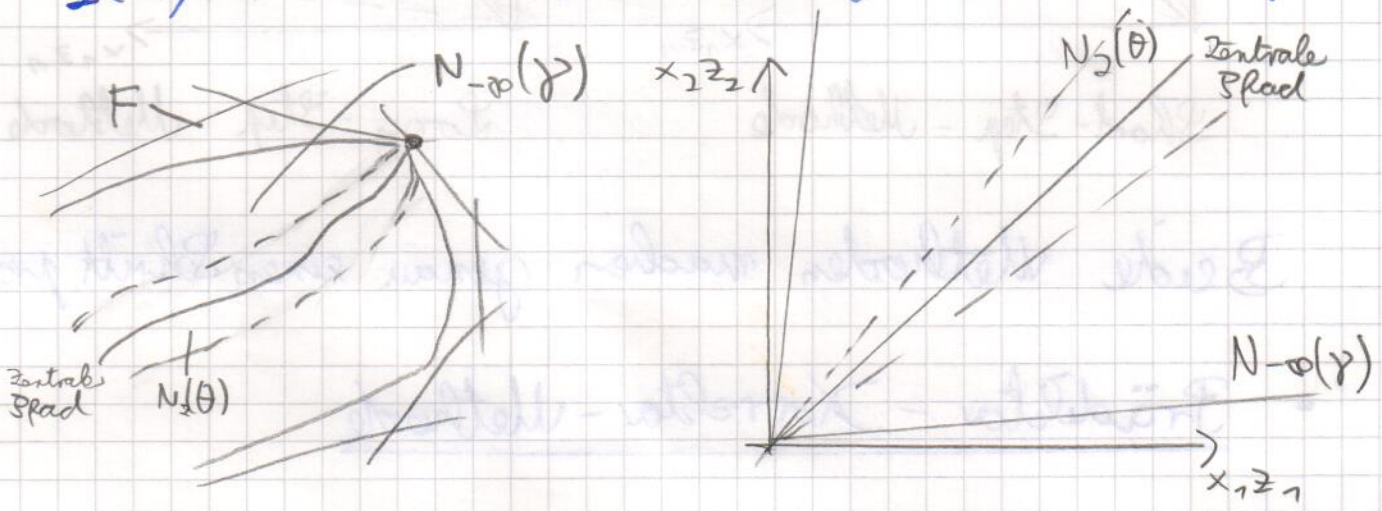
- **Entfernung vom Rand**

$$N_{-\infty}(\gamma) = \left\{ \begin{pmatrix} x \\ y \\ z \end{pmatrix} : x_i z_i \geq \gamma \cdot \mu \right\}$$

$0 < \gamma < 1$

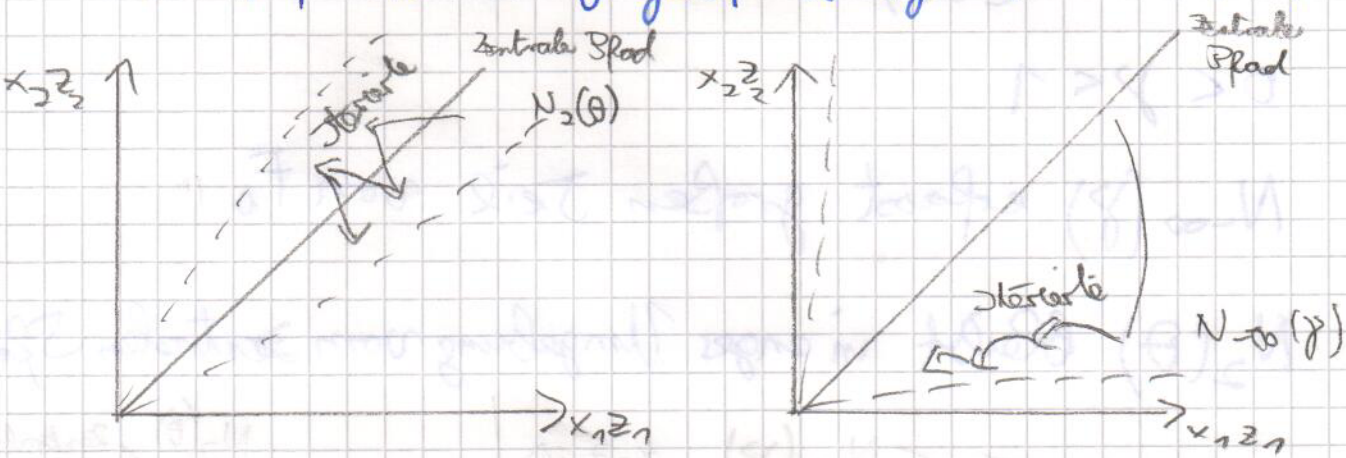
$N_{-\infty}(\gamma)$ erfasst großen Teil von F_0

$N_2(\theta)$ bleibt in enger Umgebung vom zentralen Pfad



Varianten von Innere-Punkte-Verfahren

- **Short-Step**: Wähle σ nahe bei 1, bleibt nahe beim zentralen Pfad, reduziert μ nur langsam, also Schrittweite kann immer $\alpha = 1$ gewählt werden.
- **Long-Step**: Wähle σ nahe bei 0, macht lange Schritte zur Reduzierung von μ, Wichtig: Steuerung von α, damit die Iterierten in F_0 bleiben, Zentrierung geht weitgehend verloren



Short-Step-Methode Long-Step-Methode

Beide Methoden machen genau einen Schritt pro t.

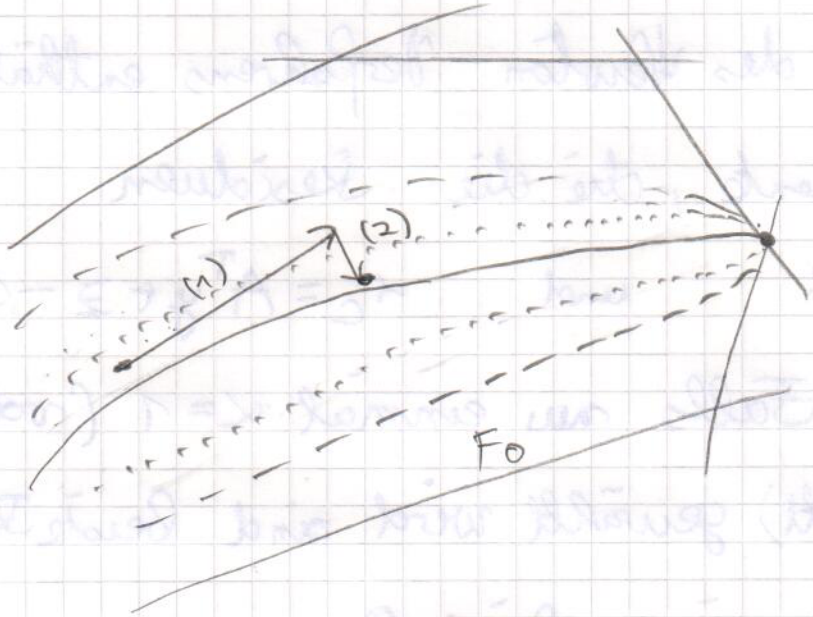
- **Prädiktor-Korrektor-Methode**

Idee: Abwechselnd

- ein Tangentialschritt
- gefolgt von einem Zentrierungsschritt

Algorithmus 10.11: PC-Algorithmus (Wright)

- Startpunkt $(x^0, y^0, z^0) \in N_2\left(\frac{1}{4}\right) \subset F_0$
- Für $k = 0, 1, 2, 3, \ldots$:
  - k gerade: wähle $\sigma_k = 0$ (Tangentialschritt)
    Schrittweite $\alpha > 0$, dass $(x^{k+1}, y^{k+1}, z^{k+1}) \in N_2\left(\frac{1}{2}\right) \subset F_0$
    (Prädiktor)
  - k ungerade: wähle $\sigma_k = 1$ (Zentrierungsschritt)
    Schrittweite $\alpha = 1$ in der Regel ok für $(x^{k+1}, y^{k+1}, z^{k+1}) \in N_2\left(\frac{1}{4}\right) \subset F_0$
    (Korrektor)



– – – $N_2\left(\frac{1}{2}\right)$

· · · $N_2\left(\frac{1}{4}\right)$

——— zentraler Pfad

(1) Tangentialschritt

(2) Zentrierungsschritt

Alle zwei Schritte ist μ reduziert und $x^{k+2} \in N_2\left(\frac{1}{4}\right)$.

Satz 10.12:

Algorithmus 10.11 ist beweisbar konvergent.

Beweis:

Stephen Wright: Primal-Dual-Interior Point Methods, Siam, 1997. ■

Zulässigkeit des Startpunkts

Wir wissen: Wenn (x^0, y^0, z^0) die linearen Gleichungen $Ax = b$, $A^T y + z = c$ erfüllt, bleiben diese für alle Iterierten (x^k, y^k, z^k) erfüllt.

Annahme: (x^0, y^0, z^0) erfüllt die linearen Gleichungen nicht.

Jedes Schritt des Newton-Verfahrens enthält eine Komponente, die die Residuen

$r_b = Ax - b$ und $r_c = A^T y + z - c$

reduziert. Falls nur einmal $\alpha = 1$ (voller Newton-Schritt) gewählt wird sind beide Residuen anschließend immer gleich 0.

Lösung des linearen Problems

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

$$\begin{pmatrix} r_b \\ r_c \\ XZe-\tau e \end{pmatrix} + J \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta z \end{pmatrix} = 0$$

Wenn $\operatorname{Rang} A = m$, ist J für alle $x, z > 0$ regulär.

Varianten zur numerischen Auflösung

1.) Reduziertes KKT-System

Löse die 3. Gleichung nach Δz auf

$$\Delta z = -X^{-1} Z \Delta x - z + \tau X^{-1} e$$

und setze in die beiden anderen Gleichungen ein:

$$\begin{pmatrix} A & 0 \\ -X^{-1}Z & A^T \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} + \begin{pmatrix} Ax-b \\ (A^T y+z-c)+z+\tau X^{-1} e \end{pmatrix} = 0$$

Dabei bleibt eine dünn besetzte Struktur von A erhalten.
Das System ist indefinit, gute Löser dafür fehlen noch.
Geeignet für iterative Methoden.

2.) Normalgleichungen

Definiere $\Theta := X^{-1}Z$ und eliminiere Δz und

$\Delta x = \Theta^{-1}(A^T \Delta y + (A^T y + z - c) - z + \tau X^{-1} e)$, bleibt

$\underbrace{A\Theta^{-1}A^T}_{=:M} \Delta y = -(Ax - b) - A\Theta^{-1}((A^T y + z - c) - z + \tau X^{-1} e)$

M ist symmetrisch und positiv definit, daher kann man zur Zerlegung Choleskys anwenden. Evtl. Dünnbesetztheit von A geht weitgehend verloren.

Endliche, polynomiale Laufzeit:

Satz 10.13:

Die primal-duale Innere-Punkte-Methode kann ein LP in endliche, polynomiale Laufzeit lösen.

Beweis: Buch von S. Wright.

11. Modellierungsbeispiele und Anwendungsprobleme

11.1 Cargo-Plane-Beispiel

Frachtflugzeug hat drei Abteile:

	Gewichtskapazität	Platzkapazität
vorne	10 t	6800 m^3
mitte	16 t	8700 m^3
hinten	8 t	5300 m^3

Die Verteilung des Gewichts in den drei Abteilen muss im gleichen Verhältnis sein wie die Kapazitäten.

Fracht, die transportiert werden soll:

	Gewicht	Volumen	Profit
C1	18 t	480 $\frac{m^3}{t}$	310 €/t
C2	15 t	650 $\frac{m^3}{t}$	330 €/t
C3	23 t	580 $\frac{m^3}{t}$	350 €/t
C4	12 t	390 $\frac{m^3}{t}$	285 €/t

Wie kann die Fracht ins Flugzeug geladen werden, um maximalen Profit zu erzielen?

Formulierung als lineares Optimierungproblem

Variablen: x_{ij} : Tonnen der Fracht i ($i = 1, \dots, 4$), die in Abteil j ($j = 1$ vorne, $j = 2$ mitte, $j = 3$ hinten) geladen werden

$x_{ij} \geq 0$

$i = 1, \dots, 4$

$j = 1, \dots, 3$

Nebenbedingungen:

- Es kann nur die vorhandene Fracht verladen werden:

$$x_{11} + x_{12} + x_{13} \leq 18$$

$$x_{21} + x_{22} + x_{23} \leq 15$$

$$x_{31} + x_{32} + x_{33} \leq 28$$

$$x_{41} + x_{42} + x_{43} \leq 12$$

- Einhaltung der Gewichtskapazitäten der Abteile:

$$x_{11} + x_{21} + x_{31} + x_{41} \leq 10$$

$$x_{12} + x_{22} + x_{32} + x_{42} \leq 16$$

$$x_{13} + x_{23} + x_{33} + x_{43} \leq 8$$

- Einhaltung der Volumenkapazitäten:

$$480 x_{11} + 650 x_{12} + 580 x_{31} + 390 x_{41} \leq 6800$$

$$480 x_{12} + 650 x_{22} + 580 x_{32} + 390 x_{42} \leq 8700$$

$$480 x_{13} + 650 x_{23} + 580 x_{33} + 390 x_{43} \leq 5300$$

- Verhältnisse des Gewichts pro Obst:

$$\frac{(x_{11} + x_{21} + x_{31} + x_{41})}{10} = \frac{(x_{21} + x_{22} + x_{23} + x_{24})}{16}$$

$$\frac{(x_{12} + x_{22} + x_{32} + x_{42})}{16} = \frac{(x_{13} + x_{23} + x_{33} + x_{34})}{8}$$

Zielfunktion:

$$\begin{aligned} \max\ & 370(x_{11} + x_{12} + x_{13}) \\ & + 380(x_{21} + x_{22} + x_{23}) \\ & + 350(x_{31} + x_{32} + x_{33}) \\ & + 285(x_{41} + x_{42} + x_{43}) \end{aligned}$$

Lösung mit GLPSOL

Diskussion:

Modellierungsannahmen:

- Die Fracht kann in beliebige Teile aufgeteilt werden.
- Die Fracht kann beliebig auf die Abteile verteilt werden.
- Die Form der Fracht spielt beim Verladen keine Rolle.
- Alle Zahlenangaben sind genau.

Sind diese Annahmen nicht erfüllt, reicht es nicht, ein LP zu formulieren, es müssen auch kombinatorische / ganzzahlige und stochastische Aspekte berücksichtigt werden.

```
\* Problem: Cargo Plane *\

Maximize
 Profit: + 310 c1f + 310 c1c + 310 c1r
         + 380 c2f + 380 c2c + 380 c2r
         + 350 c3f + 350 c3c + 350 c3r
         + 285 c4f + 285 c4c + 285 c4r

Subject To
 Weight_1: + c1f + c1c + c1r <= 18
 Weight_2: + c2f + c2c + c2r <= 15
 Weight_3: + c3f + c3c + c3r <= 23
 Weight_4: + c4f + c4c + c4r <= 12

 Weight_Front:  + c1f + c2f + c3f + c4f <= 10
 Weight_Centre: + c1c + c2c + c3c + c4c <= 16
 Weight_Rear:   + c1r + c2r + c3r + c4r <= 8

 Space_Front:  + 480 c1f + 650 c2f + 580 c3f + 390 c4f <= 6800
 Space_Centre: + 480 c1c + 650 c2c + 580 c3c + 390 c4c <= 8700
 Space_Rear:   + 480 c1r + 650 c2r + 580 c3r + 390 c4r <= 5300

 Weight_Ratio_fc: + 16 c1f + 16 c2f + 16 c3f + 16 c4f
                  - 10 c1c - 10 c2c - 10 c3c - 10 c4c = 0
 Weight_Ratio_fr: + 8 c1f + 8 c2f + 8 c3f + 8 c4f
                  - 10 c1r - 10 c2r - 10 c3r - 10 c4r = 0
 Weight_Ratio_cr: + 8 c1c + 8 c2c + 8 c3c + 8 c4c
                  - 16 c1r - 16 c2r - 16 c3r - 16 c4r = 0

Bounds

End



GLPSOL: GLPK LP/MIP Solver, v4.57
Parameter(s) specified in the command line:
 --lp Cargo_Plane.lp --output out.txt
Reading problem data from 'Cargo_Plane.lp'...
13 rows, 12 columns, 60 non-zeros
32 lines were read
GLPK Simplex Optimizer, v4.57
13 rows, 12 columns, 60 non-zeros
Preprocessing...
13 rows, 12 columns, 60 non-zeros
Scaling...
 A: min|aij| =  1.000e+00  max|aij| =  6.500e+02  ratio =  6.500e+02
GM: min|aij| =  8.006e-01  max|aij| =  1.249e+00  ratio =  1.560e+00
EQ: min|aij| =  6.410e-01  max|aij| =  1.000e+00  ratio =  1.560e+00
Constructing initial basis...
Size of triangular part is 12
*     0: obj =  -0.000000000e+00 inf =   0.000e+00 (10)
*     8: obj =   1.215157895e+04 inf =   1.332e-15 (0)
OPTIMAL LP SOLUTION FOUND
Time used:   0.0 secs
Memory used: 0.0 Mb (39701 bytes)
Writing basic solution to 'out.txt'...
```

```
Problem:
Rows:       13
Columns:    12
Non-zeros:  60
Status:     OPTIMAL
Objective:  Profit = 12151.57895 (MAXimum)
```

No.	Row name	St	Activity	Lower bound	Upper bound	Marginal
1	Weight_1	B	0		18	
2	Weight_2	NU	15		15	30
3	Weight_3	B	15.9474		23	
4	Weight_4	B	3.05263		12	
5	Weight_Front	B	10		10	
6	Weight_Centre	NU	16		16	545.329
7	Weight_Rear	B	8		8	
8	Space_Front	B	6290		6800	
9	Space_Centre	NU	8700		8700	0.342105
10	Space_Rear	B	5200		5300	
11	Weight_Ratio_fc	NS	0	0	=	39.375
12	Weight_Ratio_fr	NS	0	0	=	-35
13	Weight_Ratio_cr	B	0	0	=	

No.	Column name	St	Activity	Lower bound	Upper bound	Marginal
1	c1f	NL	0	0		-40
2	c1c	NL	0	0		-5.78947
3	c1r	NL	0	0		-40
4	c2f	B	7	0		
5	c2c	NL	0	0		-23.9474
6	c2r	B	8	0		
7	c3f	B	3	0		
8	c3c	B	12.9474	0		
9	c3r	NL	0	0		< eps
10	c4f	NL	0	0		-65
11	c4c	B	3.05263	0		
12	c4r	NL	0	0		-65

```
Karush-Kuhn-Tucker optimality conditions:

KKT.PE: max.abs.err = 1.42e-14 on row 12
        max.rel.err = 8.83e-17 on row 12
        High quality

KKT.PB: max.abs.err = 1.42e-14 on row 13
        max.rel.err = 1.42e-14 on row 13
        High quality

KKT.DE: max.abs.err = 1.14e-13 on column 4
        max.rel.err = 9.02e-17 on column 7
        High quality

KKT.DB: max.abs.err = 5.68e-14 on column 9
        max.rel.err = 5.68e-14 on column 9
        High quality

End of output
```

11.2 Das Transportproblem

Aufgabe: Transport einheitlicher Objekte von mehreren Angebots- zu mehreren Nachfrageorten, Kosten minimal

Varianten:

- ohne Kapazitätsbeschränkungen auf den Transportwegen: klassisches Transportproblem
- mit Kapazitätsbeschränkungen: kapazitiertes Transportproblem
- mit Umladeorten: Umladeproblem
- Sonderfall: Zuordnungsproblem: pro Ort nur eine Angebots- bzw. Nachfrageeinheit

Klassisches Transportproblem

Daten: m Angebotsorte A_i stellen Mengen a_i des Gutes bereit $a_i \geq 0$

n Nachfrageorte B_j fragen Mengen b_j des Gutes nach $b_j \geq 0$

Die Mengen müssen ≥ 0 sein, das Gesamtangebot muss gleich der Gesamtnachfrage sein (ausgeglichenes Transportproblem; alternativ: fiktiver Angebots- oder Nachfrageort, der die Überschüsse bereitstellt oder aufnimmt) $\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j$

Kosten $c_{ij} \geq 0$ für den Transport einer Einheit von A_i nach B_j, für fiktive Orte $c_{ij} = 0$, kann nicht transportiert werden: $c_{ij} = \infty$.

Formulierung als lineares Optimierungsproblem

Variablen: x_{ij} ist die Transportmenge von A_i nach B_j

Gesamtkosten: Summe der Kosten aller Transportwege:

$$\sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} \cdot x_{ij} = z$$ Zielfunktion

Beschränkungen:

- Angebot bei A_i wird komplett abgerufen:

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

- Nachfrage bei B_j wird komplett bedient:

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

Oft: Ganzzahligkeit der Transportmengen:

$$x_{ij} \in \mathbb{N}_0$$

Klassisches Transportproblem

$$\min \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} \cdot x_{ij}$$

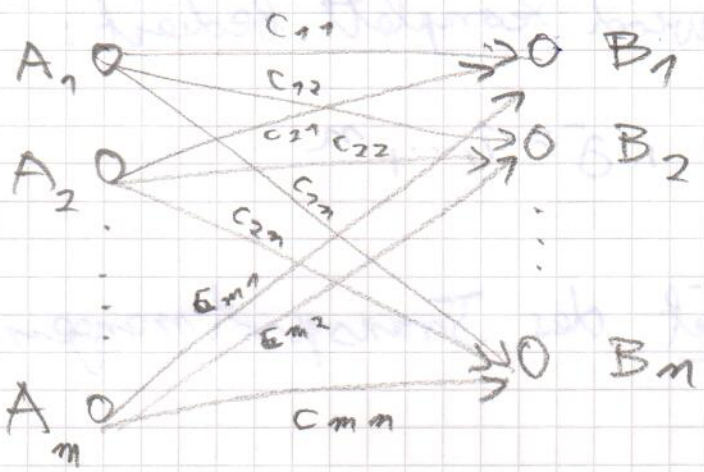
$$\sum_{j=1}^{n} x_{ij} = a_i \qquad i = 1, \dots, m$$

$$\sum_{i=1}^{m} x_{ij} = b_j \qquad j = 1, \dots, n$$

$$x_{ij} \geq 0$$

Satz:

Seien $a_i \geq 0$, $b_j \geq 0$, $c_{ij} \geq 0$, $c_{ij} < \infty$, $\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j$. Dann existiert mindestens eine Lösung für das Transportproblem.



Beschreibung durch Graph

Beispiel: Getränkeproduzent

30 Ladungen eines Getränks liegen an den Produktionsstätten Hamburg (16 Ladungen) und Paris (14 Ladungen) auf Lager.

15 Ladungen sollen nach Lissabon, 5 nach Barcelona und 10 nach Florenz geliefert werden. Transportkosten pro Ladung in TE:

von/nach	Lissabon	Barcelona	Florenz	Angebot
Hamburg	6	4	3	16 TL
Paris	3	2	2	14 TL
Nachfrage	15 TL	5 TL	10 TL	30 TL

Variablen x_{11}, x_{12}, x_{13}

x_{21}, x_{22}, x_{23}

LP-Formulierung:

$$\min 6x_{11} + 4x_{12} + 3x_{13} + 3x_{21} + 2x_{22} + 2x_{23}$$

$$x_{11} + x_{12} + x_{13} = 16 \quad \text{(Angebot)}$$

$$x_{21} + x_{22} + x_{23} = 14$$

$$x_{11} + x_{21} = 15$$

$$x_{12} + x_{22} = 5 \quad \text{(Nachfrage)}$$

$$x_{13} + x_{23} = 10$$

$$x_{11}, \dots, x_{23} \geq 0$$

Lösung mit GLPSOL: $z^* = 98$

$x_{11} = 1 \quad x_{12} = 5 \quad x_{13} = 10$

$x_{21} = 14 \quad x_{22} = 0 \quad x_{23} = 0$

Beobachtung: Die Lösungsvariablen haben ganzzahlige Werte. Das gilt allgemein:

Satz:

Seien im klassischen Transportproblem die Angebots- und Nachfragemengen ganzzahlig. Sofern eine zu-

lässige Lösung existiert, liefert das Simplex-Verfahren immer eine ganzzahlige Lösung.

Beweis:

Die Nebenbedingungsmatrix ist „vollständig unimodular". Daher ist jede Ecke des Polyeders der zulässigen Punkte ganzzahlig. ■

Definition:

Eine Matrix heißt vollständig (total) unimodular, falls ihre Minoren (Unterdeterminanten) nur Werte 1, 0, -1 annehmen.

Satz:

Die Matrix der Nebenbedingungen des klassischen Transportproblems (mit ganzzahligen Angebots- und Nachfragemengen) ist vollständig unimodular. In jeder Spalte stehen höchstens zwei Einsen. Jede Basismatrix hat Dreiecksgestalt.

Folgerung:

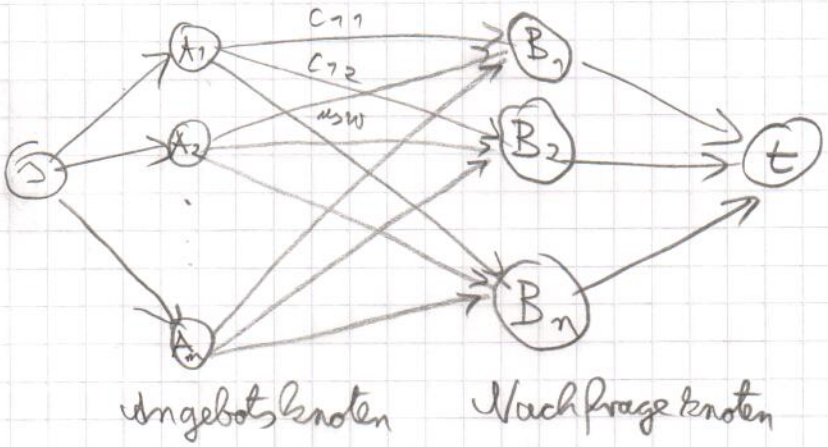
Ist die rechte Seite der Weben bedingungen ganzzahlig, so ist jede Basislösung ebenfalls ganzzahlig. Also ist jede Ecke des Polyeders der zulässigen Punkte ganzzahlig.

Anfangsbasislösung:

Statt des normalen Phase I gibt es verschiedene spezielle Ansätze:

- Nord-West-Ecken-Regel
- Minimalkostenregel

Formulierung als Min-Cost-Flow-Problem



Zusätzliche Knoten:

s: Quelle

t: Senke

Flussproblem: Bestimme den kostenminimalen Fluss von s nach t.

Neben dem LP-Lösungsansatz können min-cost-flow-Probleme auch mit effizienten Graphenalgorithmen der kombinatorischen Optimierung gelöst werden.

```
\* Problem: Getraenketransport *\

Minimize
  Kosten: 6 x11 + 4 x12 + 3 x13 + 3 x21 + 2 x22 + 2 x23

Subject To
  Angebot1: x11 + x12 + x13 = 16
  Angebot2: x21 + x22 + x23 = 14
  Nachfrage1: x11 + x21 = 15
  Nachfrage2: x12 + x22 = 5
  Nachfrage3: x13 + x23 = 10

Bounds

Generals

Binaries

End
```

```
GLPSOL: GLPK LP/MIP Solver, v4.57
Parameter(s) specified in the command line:
 --lp Getraenketransport.lp --output out.txt
Reading problem data from 'Getraenketransport.lp'...
5 rows, 6 columns, 12 non-zeros
19 lines were read
GLPK Simplex Optimizer, v4.57
5 rows, 6 columns, 12 non-zeros
Preprocessing...
5 rows, 6 columns, 12 non-zeros
Scaling...
 A: min|aij| =  1.000e+00  max|aij| =  1.000e+00  ratio =  1.000e+00
Problem data seem to be well scaled
Constructing initial basis...
Size of triangular part is 4
      0: obj =   9.100000000e+01 inf =   6.000e+00 (1)
      1: obj =   1.030000000e+02 inf =   0.000e+00 (0)
*     2: obj =   9.800000000e+01 inf =   0.000e+00 (0)
OPTIMAL LP SOLUTION FOUND
Time used:   0.0 secs
Memory used: 0.0 Mb (39701 bytes)
Writing basic solution to 'out.txt'...
```

```
Problem:
Rows:       5
Columns:    6
Non-zeros:  12
Status:     OPTIMAL
Objective:  Kosten = 98 (MINimum)
```

No.	Row name	St	Activity	Lower bound	Upper bound	Marginal
1	Angebot1	NS	16	16	=	3
2	Angebot2	NS	14	14	=	< eps
3	Nachfrage1	NS	15	15	=	3
4	Nachfrage2	NS	5	5	=	1
5	Nachfrage3	B	10	10	=	

No.	Column name	St	Activity	Lower bound	Upper bound	Marginal
1	x11	B	1	0		
2	x12	B	5	0		
3	x13	B	10	0		
4	x21	B	14	0		
5	x22	NL	0	0		1
6	x23	NL	0	0		2

Lineare Optimierung mit CPLEX

Testbeispiel: Das Orangenproblem

$$\begin{aligned} \max\ & 0.2x_1 + 0.25x_2 \\ & 2x_1 + x_2 \leq 200 \\ & 0.2x_1 + x_2 \leq 110 \\ & x_1, x_2 \geq 0 \end{aligned}$$

x_1 produzierte Menge Orangensaft, x_2: produzierte Menge Orangenmarmelade

Aufruf aus der Kommandozeile

lp-File: Orangen.lp

```
\* Orangenproblem *\

Maximize
  Gewinn: 0.2 x1 + 0.25 x2

Subject To
  Orangen: 2 x1 + 1 x2 <= 200
  Zucker: 0.2 x1 + 1 x2 <= 110

End
```

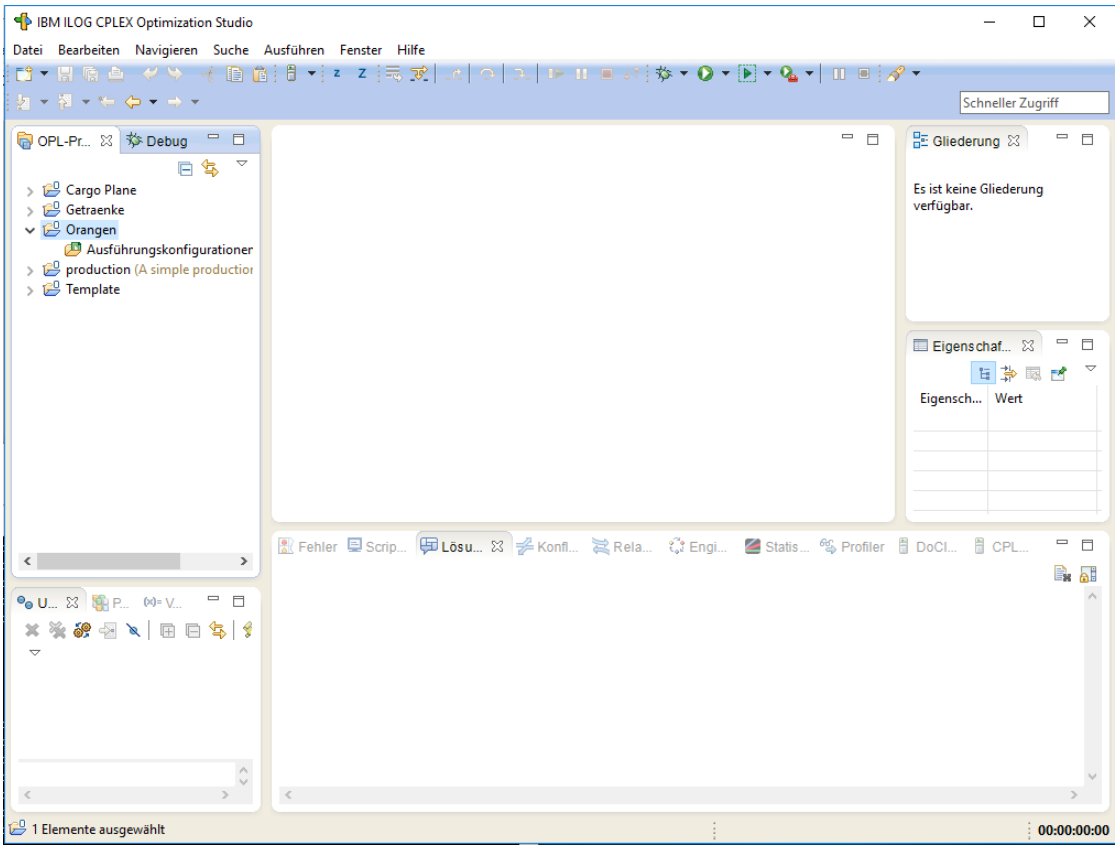
Kommandoskript: Orangen.cmd

```
read Orangen.lp
optimize
display solution variables -
display solution dual -
```

Aufruf aus der Kommandozeile

```
cplex -f Orangen.cmd
```

Aufruf aus dem CPLEX Optimization Studio



- OPL-Projekt Orangen anlegen
- Datei Orangen.mod hinzufügen durch Drag & Drop aus dem Windows Explorer
- doppelklicken zum editieren
- zu Ausführungskonfiguration hinzufügen
- Ausführungskonfiguration ausführen

Direkte Modellierung: Orangen.mod

```
/*********************************************
 * OPL 12.6.3.0 Model
```

```
 * Author: skoerkel
 * Creation Date: 18.01.2017 at 20:15:48
 *********************************************/

dvar float+ Orangensaft;
dvar float+ Orangenmarmelade;

maximize
   0.2 * Orangensaft + 0.25 * Orangenmarmelade;

subject to
{  Orangen: 2 * Orangensaft + Orangenmarmelade <= 200;
   Zucker: 0.2 * Orangensaft + Orangenmarmelade <= 110;
}

execute
{  writeln("Zielfunktionswert = ", cplex.getObjValue() );
   writeln("Orangensaft = ", Orangensaft );
   writeln("Orangenmarmelade = ", Orangenmarmelade );
   writeln("Orangen dual = ", Orangen.dual );
   writeln("Zucker dual = ", Zucker.dual );
}
```

Modellierung mit Daten: lp.mod

```
/*********************************************
 * OPL 12.6.3.0 Model
 * Author: skoerkel
 * Creation Date: 18.01.2017 at 20:30:25
 *********************************************/

int m = ...;
int n = ...;

range Rows = 1..m;
range Cols = 1..n;

float A[Rows][Cols] = ...;
float b[Rows] = ...;
float c[Cols] = ...;
```

```
dvar float+ x[Cols];

maximize
   sum( j in Cols ) c[j] * x[j];

subject to
{  forall( i in Rows )
      ct: sum( j in Cols ) A[i][j] * x[j] <= b[i];
}

execute
{  writeln("Zielfunktionswert = ", cplex.getObjValue() );
   write("Primale Loesung: x = [ " );
   for( var j in Cols ) write( x[j], " " );
   writeln("]");
   write("Duale Loesung: u = [ " );
   for( var i in Rows ) write( ct[i].dual, " " );
   writeln("]");
}
```

Datenfile: Orangen.dat

```
/*********************************************
 * OPL 12.6.3.0 Data
 * Author: skoerkel
 * Creation Date: 18.01.2017 at 20:50:25
 *********************************************/

m = 2;
n = 2;

A = [ [ 2 1 ]
      [ 0.2 1 ] ];

b = [ 200 110 ];

c = [ 0.2 0.25 ];
```

- lp.mod und Orangen.dat zum OPL-Projekt Orangen hinzufügen

- beide zu einer neuen Ausführungskonfiguration hinzufügen
- diese Ausführungskonfiguration ausführen

Anderes Datenfile: diet.dat

Anwendung auf weiteres Beispiel Diet

- diet.dat zum OPL-Projekt Orangen hinzufügen
- lp.mod und diet.dat zu einer neuen Ausführungskonfiguration hinzufügen
- diese Ausführungskonfiguration ausführen

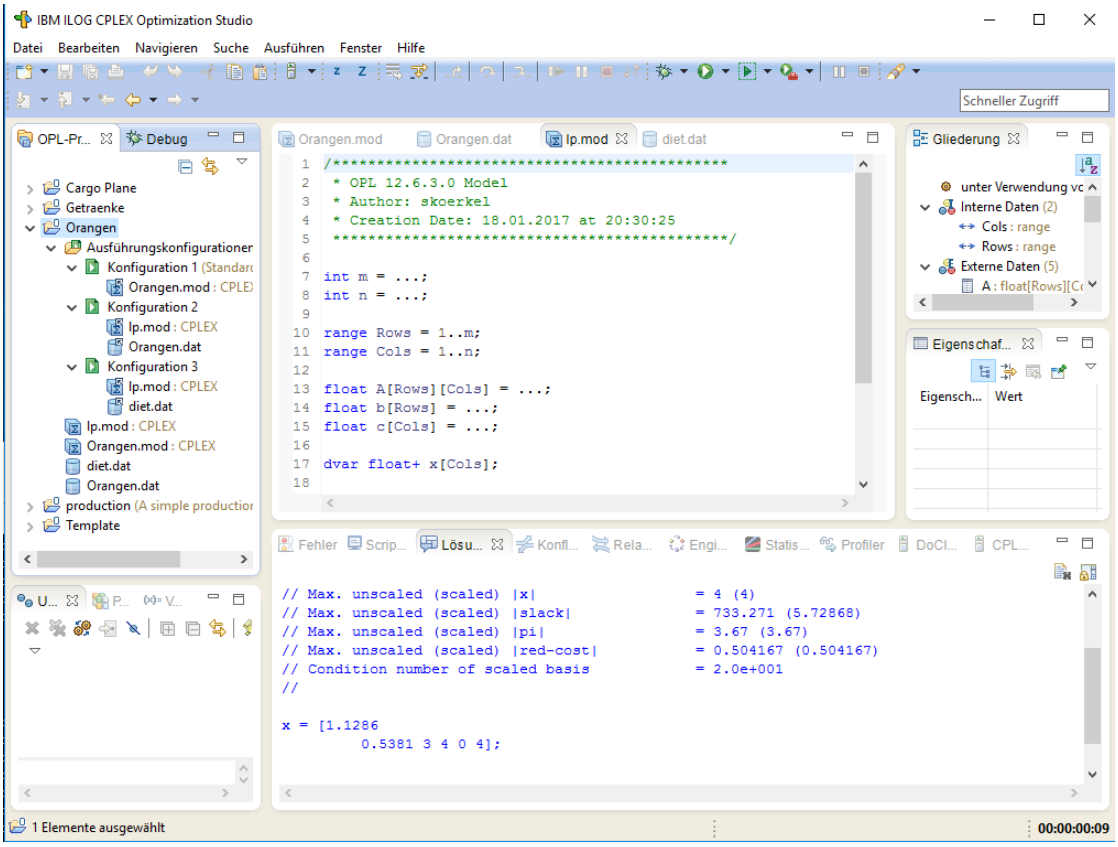
```
/*********************************************
 * OPL 12.6.3.0 Data
 * Author: skoerkel
 * Creation Date: 18.01.2017 at 18:29:35
 *********************************************/

m = 14;
n = 6;

A = [ [ -24, -8, -3, -5, -8.5, -2 ]
      [ -3, -3, -3, -3, 0, -1 ]
      [ -44, -96, -5, -27, -13, -27 ]
      [ 530, 510, 25, 140, 130, 110 ]
      [ -530, -510, -25, -140, -130, -110 ]
      [ 65, 30, 0, 0, 20, 0 ]
      [ -10, -3, -27, 0, -10, -2 ]
      [ -4, -33, -137, 0, -4, -100 ]
      [ 10, 8, 0, 0, 3, 0 ]
      [ 1020, 180, 24, 270, 125, 0 ]
      [ 0, 0, 0, 0, 1, 1 ]
      [ 0, 0, 1, 0, 0, 0 ]
      [ 0, 0, 0, 1, 0, 0 ]
      [ 0, 0, 0, 0, 0, 1 ] ];

b = [ -56 -30 -130 2200 -1800 300 -100 -100 24 2400 6 3 4 4 ];

c = [ -3.29 -3.99 -0.45 -0.48 -0.4 -0.25 ];
```



Literatur

T. Flatberg. *A short OPL tutorial.*
`http://home.agh.edu.pl/~cholda/wp-content/uploads/2012/09/opl_tutorial.pdf`.