Bemerkung

Wesentlich für die Anwendbarkeit der linearen Gaußschen Ausgleichsrechnung ist, daß für die zu bestimmenden Größen eine lineare Beziehung gegeben ist, z. B. $y(x) = a + bx$.

Ist die gegebene Beziehung (etwa aus physikalischen Gründen) nichtlinear, so kann man versuchen, aus ihr eine lineare Beziehung für unter Umständen andere Größen zu gewinnen, aus denen sich dann nachträglich die eigentlich gesuchten Größen bestimmen lassen.

Beispiel E.3

$$y(x) = \frac{a}{1+bx} \Longrightarrow \frac{1}{a} + \frac{b}{a}x = \frac{1}{y(x)} \; = \; \tilde{y} = \tilde{a} + \tilde{b}x$$

Lineare Optimierung

- lineare Optimierungsprobleme
- Polyeder
- Simplex-Algorithmus
- Dualität
- kombinatorische ganzzahlige lineare Optimierungsprobleme
- Branch & Bound
- Schnittebenenverfahren

Einführendes Beispiel: Barkeeper

Cocktails:

- Daiquiri (45 ml weißer Rum, 30 ml Cointreau, 30 ml Zitronensaft, 15 ml Zuckersirup, Eis), 5.50 Euro
- Kamikaze (30 ml Wodka, 30 ml Cointreau, 30 ml Zitronensaft, 1 Schuß Limonensirup, Eis), 4.50 Euro
- Long Island Ice Tea (20 ml Wodka, 20 ml weißer Rum, 20 ml Gin, 20 ml Cointreau, 4 TL Zitronensaft, 4 TL Orangensaft, 1/8 l Cola, 1 Orangenscheibe, Eis), 7.00 Euro

Vorhandene Spirituosen: 5 l weißer Rum, 6 l Cointreau, 4 l Wodka und 3 l Gin

Welche Cocktails muß der Barkeeper mixen, um möglichst viel Geld einzunehmen?

Variablen:
x_1: Anzahl Daiquiris
x_2: Anzahl Kamikazes
x_3: Anzahl Long Island Ice Teas

Zielfunktion: Maximiere die Einnahmen:

$$\max 5.50x_1 + 4.50x_2 + 7.00x_3$$

Nebenbedingungen:

$$\begin{array}{lrcrcrcr}
\text{Weißer Rum:} & 45x_1 & & & + & 20x_3 & \le & 5000 \\
\text{Cointreau:} & 30x_1 & + & 30x_2 & + & 20x_3 & \le & 6000 \\
\text{Gin:} & & & & & 20x_3 & \le & 3000 \\
\text{Wodka:} & & & 30x_2 & + & 20x_3 & \le & 4000
\end{array}$$

Optimierungsproblem:

$$\max \begin{pmatrix} 5.50 \\ 4.50 \\ 7.00 \end{pmatrix}^T x$$

$$\begin{pmatrix} 45 & & 20 \\ 30 & 30 & 20 \\ & & 20 \\ & 30 & 20 \end{pmatrix} x \le \begin{pmatrix} 5000 \\ 6000 \\ 3000 \\ 4000 \end{pmatrix}$$

Schreibweise: $\le$ bei Vektoren $u, v \in \mathbb{R}^n$

$$u \le v :\Longleftrightarrow \forall i = 1, \dots, n : u_i \le v_i$$

($\ge, <, >$ analog)

Lösung mit MATLAB:

```
>> A = [ [ 45, 0, 20 ]; [30, 30, 20 ]; [ 0, 0, 20 ]; [ 0, 30, 20 ] ]

A =
    45     0    20
    30    30    20
     0     0    20
     0    30    20

>> b = [ 5000, 6000, 3000, 4000 ]

b =
        5000         6000         3000         4000

>> c = [- 5.5, -4.5, -7 ]

c =
   -5.5000   -4.5000   -7.0000

>> x = linprog( c, A, b )
Optimization terminated.

x =
   44.4444
   33.3333
  150.0000
```


Lineare Optimierungsprobleme

Definition E.4

Optimierungsprobleme mit linearer Zielfunktion und linearen (Gleichungs- und Ungleichungs-) Nebenbedingungen nennt man **Lineare Optimierungsprobleme**, **Lineare Programme**, **LPs**.

Allgemeinste Form:

$$\begin{array}{lllll} \multicolumn{4}{l}{\max c^T x + d^T y} & \text{Zielfunktion} \\ Ax + & By & \le & a & \le\text{-Ungleichungen} \\ Cx + & Dy & \ge & b & \ge\text{-Ungleichungen} \\ Ex + & Fy & = & g & \text{Gleichungen} \\ & x & \ge & 0 & \text{vorzeichenbeschränkte Variablen} \end{array}$$

Weiteres Beispiel: Tschebyscheffsche Approximationsaufgabe

Überbestimmtes lineares Gleichungssystem $Ax = b$, $A \in \mathbb{R}^{m \times n}$, $m > n$

Lösung mit kleinstem Fehler:

$$\min_x \|Ax - b\|_\infty$$

in der Norm $\|Ax - b\|_\infty = \max_{i=1,\dots,m} \left| \sum_{j=1}^{n} a_{ij} x_j - b_i \right|$

(siehe lineare Ausgleichsprobleme in 1)

Umformulierung: zusätzliche Variable $\delta \in \mathbb{R}$

$$\min_{x,\delta} \delta$$
$$\left| \sum_{j=1}^{n} a_{ij} x_j - b_i \right| \leq \delta$$

Auflösung der Beträge ergibt ein LP:

$$\min_{x,\delta} \delta$$
$$\sum_{j=1}^{n} a_{ij} x_j - b_i \leq \delta$$
$$\sum_{j=1}^{n} a_{ij} x_j - b_i \geq -\delta$$

Transformationen

1. min-Probleme werden zu max-Problemen, indem man die Zielfunktion mit -1 multipliziert:

$$\min c^T x \iff \max -c^T x$$

2. $\geq$-Ungleichungen werden zu $\leq$-Ungleichungen, indem man sie mit -1 multipliziert:

$$Ax \geq b \iff -Ax \leq -b$$

3. Gleichungen kann man durch Paare von Ungleichungen ersetzen:

$$Ax = b \iff \begin{pmatrix} Ax \leq b \\ Ax \geq b \end{pmatrix}$$

Transformationen

4. Ungleichungen kann man durch Einführung von Schlupfvariablen zu Gleichungen machen:

$$Ax \leq b \iff \begin{pmatrix} Ax + s = b \\ s \geq 0 \end{pmatrix}$$

5. Vorzeichenunbeschränkte Variablen kann man in Paare von vorzeichenbeschränkten Variablen aufsplitten:

$$x = y - z, \quad y \geq 0, \quad z \geq 0$$

6. Die Vorzeichenbeschränkungen kann man (formal) zu den anderen Ungleichungen hinzunehmen:

$$\begin{pmatrix} Ax \leq b \\ x \geq 0 \end{pmatrix} \iff \begin{pmatrix} A \\ -\mathbb{1} \end{pmatrix} x \leq \begin{pmatrix} b \\ 0 \end{pmatrix}$$

Folgerung E.5

Man kann jedes allgemeine LP in der **Standardform**

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

oder in der Form

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

schreiben.

Bemerkung E.6

Natürlich kann man auch Nebenbedingungen und Variablen **skalieren**.
Das ist wichtig bei der numerischen Behandlung.

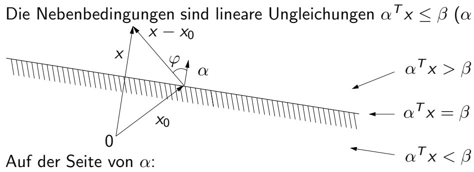
Wir betrachten im folgenden lineare Programme der (allgemeinen) Form

$$\begin{array}{l} \max c^T x \\ Ax \leq b \end{array} \qquad \text{(P)}$$

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

Geometrische Untersuchung

Die Nebenbedingungen sind lineare Ungleichungen $\alpha^T x \leq \beta$ $(\alpha \neq 0)$



Auf der Seite von α:

$$0 < \cos\varphi = < \alpha, x - x_0 > = \alpha^T(x - x_0) = \alpha^T x - \alpha^T x_0 = \alpha^T x - \beta$$

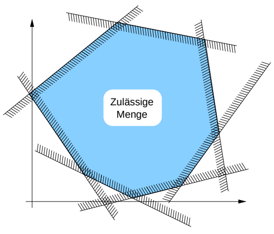
Auf der Seite von $-\alpha$: $\quad \alpha^T x < \beta$

$\{x : \alpha^T x = \beta\}$ ist eine **Hyperebene**.

$\{x : \alpha^T x \leq \beta\}$ (oder $\geq$) ist ein **Halbraum**.

Das System $Ax \leq b$ besteht aus den Ungleichungen

$$\begin{array}{rcccl} \alpha_1^T x & := & \sum_{j=1}^n a_{1j} x_j & \leq & b_1 \\ & & \vdots & & \\ \alpha_m^T x & := & \sum_{j=1}^n a_{mj} x_j & \leq & b_m \end{array} \qquad A = \begin{pmatrix} \alpha_1^T \\ \vdots \\ \alpha_m^T \end{pmatrix}$$



Polyeder

Jede Zeile des Ungleichungssystems beschreibt einen Halbraum. Die zulässige Menge ist der Durchschnitt von (endlich vielen) Halbräumen. Dies nennt man ein **Polyeder** (wenn beschränkt auch **Polytop**).

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

Annahme: Das Polyeder ist voll-(n)-dimensional.

($\dim P = n - \mathrm{Rang} A_{\mathrm{eq}(P)}$, wobei $A_{\mathrm{eq}(P)}$ die Teilmatrix von A zu den Ungleichungen ist, die von allen Punkten aus P mit Gleichheit erfüllt werden.)

Konvexität und Ecken

Bemerkung E.7

Ein Polyeder ist eine konvexe Menge. **Konvex** *bedeutet, daß mit je zwei Punkten auch ihre gesamte Verbindungsstrecke in der Menge liegt:*

$$x \neq y \in P \Longrightarrow \forall \theta \in [0;1] : x(\theta) := x + \theta(y - x) \in P.$$

Ein Punkt des Polyeders, der nie im Innern, sondern immer am Rand von solchen Verbindungsstrecken liegt, heißt **Ecke***:*

$$z \text{ Ecke} \iff \forall x \neq y \in P : (\exists \theta \in [0;1] : z = x + \theta(y - x)) \Longrightarrow (\theta = 0 \vee \theta = 1)$$

Satz E.8

Sei $P := \{x : Ax \leq b\}$ ($A \in \mathbb{R}^{m \times n}$) ein volldimensionales Polyeder. Dann gilt: $x_0 \in P$ ist Ecke von P genau dann, wenn n Ungleichungen mit linear unabhängigen Zeilen von A mit Gleichheit erfüllt sind:

$$\exists B \subset \{1, \ldots, m\}, |B| = n : \forall i \in B : \alpha_i^T x = b_i, \ \{\alpha_i, i \in B\} \text{ lin. unabh.}$$

Schreibweise:

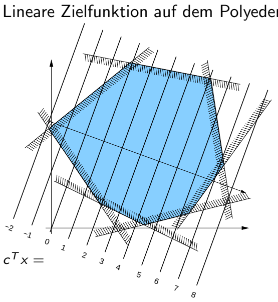
$$A_B := \begin{pmatrix} \vdots \\ \alpha_i^T \\ \vdots \end{pmatrix}_{i \in B}$$

ist eine Teilmatrix von A aus den Zeilen mit Indizes aus B, genannt **Basismatrix**, und ist invertierbar.

$$A_B x = b_B$$

Ab jetzt: Annahmen: P ist volldimensional, und P hat (mindestens) eine Ecke.

Lineare Zielfunktion auf dem Polyeder:



Beobachtung:

Satz: *Wenn das LP eine Optimallösung hat, dann gibt es auch eine optimale Ecklösung.*

Berechnung der optimalen Ecke

Idee:

1. Starte mit einer zulässigen Ecke (d. h. einer Ecke, die alle Nebenbedingungen erfüllt).
2. Gehe zur nächsten Ecke, wobei die Zielfunktion ansteigt (bzw. nicht abnimmt).
3. Tue dies, bis keine Verbesserung mehr möglich ist.
4. Vermeide, Ecken zweimal zu besuchen.

Simplex-Algorithmus der linearen Programmierung

Dantzig, 1947

hier: geometrische Version

0. **Starte mit einer zulässigen Ecke** $x^0 \in \mathbb{R}^n$.
   Sei B die zugehörige Zeilenindexmenge und A_B die zugehörige Basismatrix:
   $$A_B x^0 = b_B \text{ bzw. } x^0 = A_B^{-1} b_B$$

1. **Ist** x^0 **optimal?**
   Definiere
   $$u^T := c^T A_B^{-1} \in \mathbb{R}^n$$
   Falls $u \geq 0$, dann gilt für alle x mit $Ax \leq b$:
   $$c^T x = u^T A_B x \leq u^T b_B = u^T A_B x^0 = c^T x^0$$
   $\Longrightarrow$ x^0 ist optimal. STOP.

2. **Finde eine Richtung mit nicht abnehmender Zielfunktion.**
   Es gibt also (mindestens) ein $i_0 \in B$ mit $u_{i_0} < 0$.
   Sei
   $$d := -A_B^{-1} e_{i_0}$$
   Dann gilt für alle $\lambda \geq 0$:
   $$\begin{aligned} c^T(x^0 + \lambda \cdot d) - c^T x^0 &= \lambda \cdot c^T d = -\lambda \cdot c^T A_B^{-1} e_{i_0} \\ &= -\lambda \cdot u^T e_{i_0} = -\lambda \cdot u_{i_0} \geq 0 \end{aligned}$$
   $\Longrightarrow$ Entlang der Richtung d nimmt die Zielfunktion nicht ab.

3. **Ist das Problem unbeschränkt?**
   Finde die nächste zulässige Ecke in Richtung d, d. h.
   $$\alpha_j^T(x^0 + \lambda \cdot d) = \alpha_j^T x^0 + \lambda \cdot \alpha_j^T d \overset{!}{\leq} b_j$$
   Es gilt
   $$\alpha_j^T x^0 \leq b_j \quad \forall j \in \{1, \ldots, m\}$$
   und
   $$\alpha_j^T d = -\alpha_j^T A_B^{-1} e_{i_0} = -\delta_{j i_0} \leq 0 \quad \forall j \in B$$
   Falls auch
   $$\alpha_j^T d \leq 0 \quad \forall j \in \{1, \ldots, m\} \setminus B$$
   dann können wir jedes $\lambda > 0$ wählen und bleiben immer zulässig.
   $\Longrightarrow$ Das Problem ist unbeschränkt. STOP.

4. **Bestimme die Schrittweite, um zur nächsten zulässigen Ecke zu gehen.**
   Es gibt also (mindestens) ein $j \in \{1, \ldots, m\} \setminus B$ mit $\alpha_j^T d > 0$.
   Bedingung für λ:

$$\lambda \leq \frac{b_j - \alpha_j^T x^0}{\alpha_j^T d} \quad \forall j \in \{1, \ldots, m\} \setminus B \text{ mit } \alpha_j^T d > 0$$

   Sei

$$\lambda := \min \left\{ \frac{b_j - \alpha_j^T x^0}{\alpha_j^T d}, j \in \{1, \ldots, m\} \setminus B \text{ mit } \alpha_j^T d > 0 \right\}$$

   und $j_0 \in \{1, \ldots, m\} \setminus B$ ein zugehöriger Index.

5. **Gehe zur nächsten Ecke.**
   Definiere

$$x^1 := x^0 + \lambda \cdot d$$

   und

$$B_{neu} := B \setminus \{i_0\} \cup \{j_0\}$$

   Update von A_B und A_B^{-1}.
   Weiter mit Schritt 1 und x^1 statt x^0.

Bemerkungen zum Simplex-Algorithmus

Bei der Wahl von i_0 und j_0 hat man u. U. mehrere Möglichkeiten. Durch bestimmte Strategien („wähle den kleinsten Index“) kann man vermeiden, dieselbe Ecke mehrmals zu besuchen. Da es nur endlich viele Ecken gibt, terminiert der Algorithmus dann nach endlich vielen Iterationen.

Bemerkungen zum Simplex-Algorithmus

Die Anzahl aller Ecken ist exponentiell in m, n ($\sim \binom{m}{n}$). Man kann Beispiele konstruieren (Klee-Minty), für die der Simplex-Algorithmus alle Ecken besucht. Die Worst-Case-Laufzeit des Simplex-Algorithmus ist also nicht polynomial.

Aber: Khachian (1979) und Karmakar (1984) haben polynomiale Algorithmen für LP gefunden. Also ist $LP \in \mathcal{P}$.

In der Praxis ist der Simplex-Algorithmus sehr konkurrenzfähig, typische Laufzeit $\sim 3m$, $\sim \log n$.

Alternative: Innere-Punkt-Methoden (später in dieser Vorlesung).

Wie erhält man eine zulässige Ecke x^0, um den Simplex zu starten?

Formuliere ein Hilfsproblem, z. B. (mit $y \in \mathbb{R}$)

$$\max_{x,y} y$$
$$Ax - y \cdot b \leq 0 \quad -y \leq 0 \quad y \leq 1$$

Für dieses Problem ist der Punkt $x = 0$, $y = 0$ zulässig, er kann also als Startpunkt genommen werden, um das Hilfsproblem mit dem Simplex-Algorithmus zu lösen. (sogenannte Phase I)

In der Lösung (x^0, y^0) ist y^0 entweder 0 oder 1 (die Lösung ist eine Ecke).

Wenn $y^0 = 0$, dann ist $Ax > y \cdot b$ für alle x und alle $y > 0$, also auch für $y = 1$. Das eigentliche LP ist also unzulässig.

Falls $y^0 = 1$, dann ist $Ax^0 \leq b$, x^0 kann also als zulässige Startlösung für den eigentlichen Simplex verwendet werden.

Folgerung: Einen zulässigen Punkt zu finden, ist eine genauso schwere Aufgabe, wie einen optimalen Punkt zu finden.