Lineare und Ganzzahlige Optimierung

- Zeit: Dienstag und Donnerstag, 16:15 - 17:45
- Die Vorlesung findet vorerst als Zoom-Meeting statt.
- Homepage:
  `www.or.uni-bonn.de/lectures/ss20/lgo_ss20.html`
- Ein Skript und alle Folien finden sich auf der Homepage.
- Es wird mündliche Prüfungen geben.

Übungen

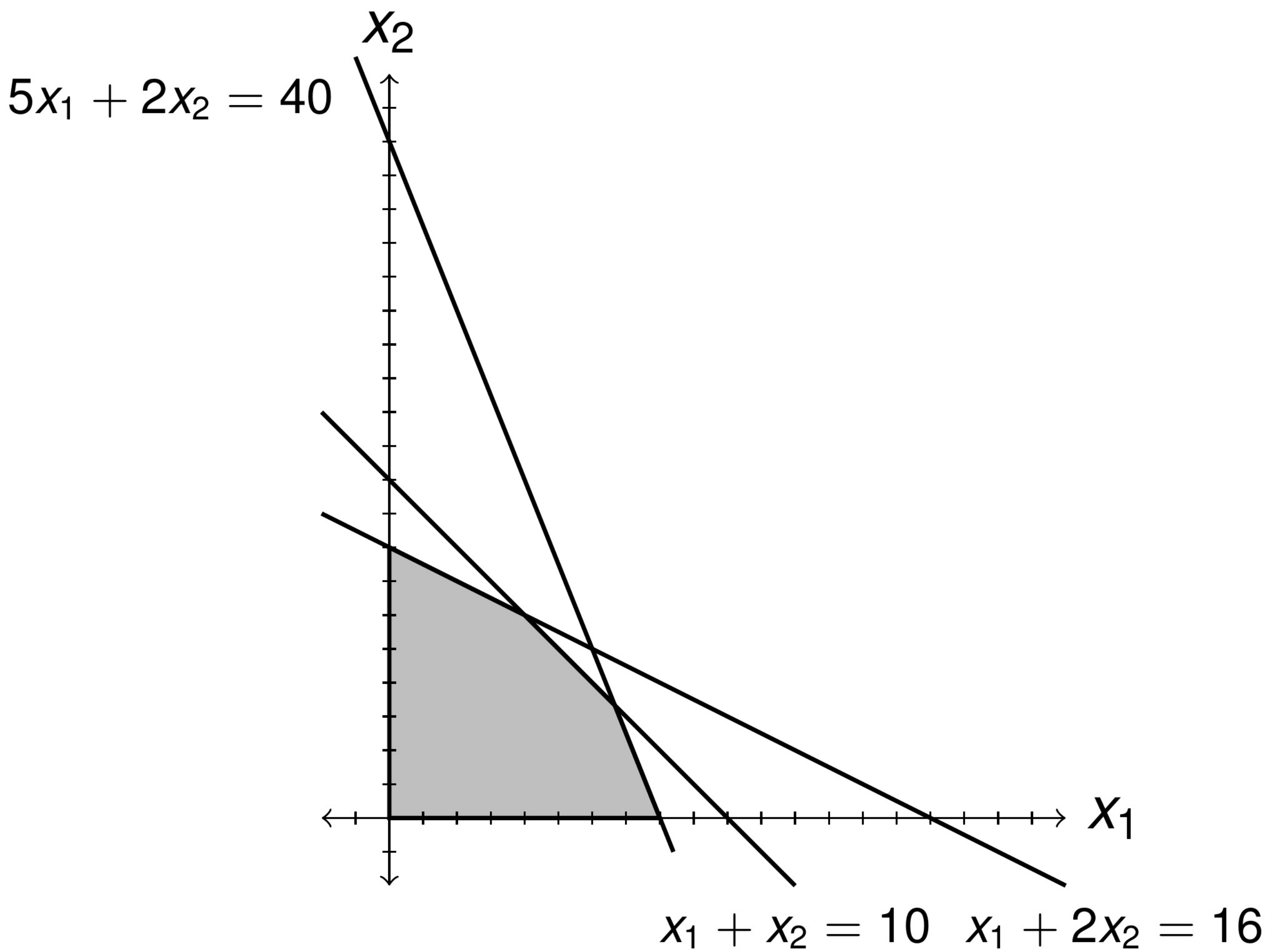
- Es gibt zweistündige Übungen (ebenfalls als Zoom-Meetings)
  - Gruppe A: Mo 14-16
  - Gruppe B: Di 14-16
- Eine Anmeldung ist noch möglich per E-Mail an `brenner@or.uni-bonn.de` (mit Gruppenwunsch).
- Jeden Donnerstag wird ein Übungszettel ausgegeben, der innerhalb von einer Woche zu bearbeiten ist.
- Es wird auch Programmierübungen geben.
- 50 % der Gesamtpunkte sind für die Zulassung zur Prüfung notwendig.
- Teilnehmer können in Zweiergruppen ihre Lösungen abgeben.
- Jeder Teilnehmer in einer Zweiergruppe muss in der Lage sein, alle Lösungen der Gruppe zu erklären.
- Die Übungen beginnen am 27. April.

Einführung

Beispiel-LP

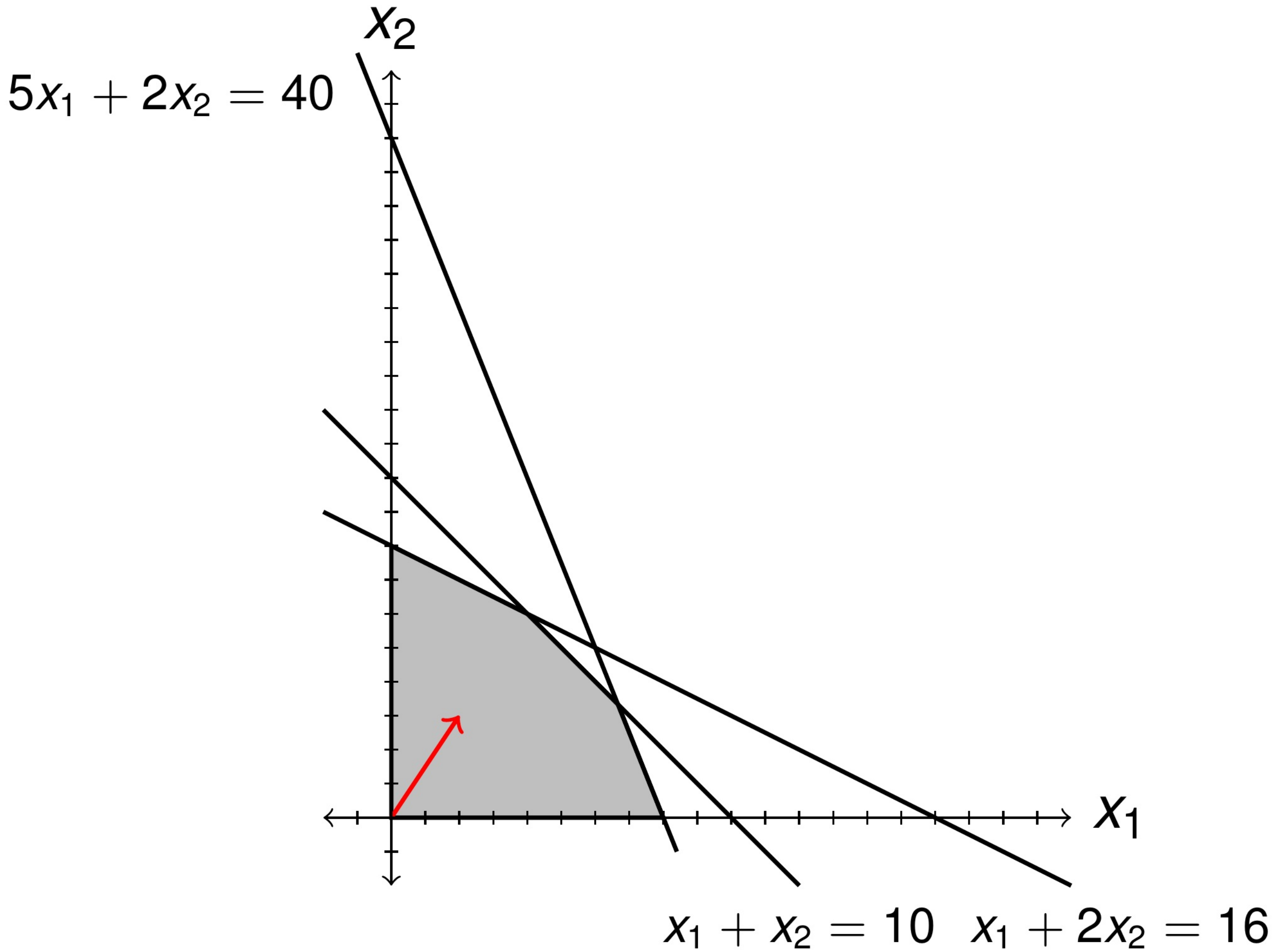
$$
\begin{array}{llcll}
\max & 2x_1 + 3x_2 & & & \text{// Lineare Zielfunktion} \\
\text{s.d.} & x_1 + x_2 & \leq & 10 & \text{//} \\
 & x_1 + 2x_2 & \leq & 16 & \text{// Lineare Nebenbedingungen} \\
 & 5x_1 + 2x_2 & \leq & 40 & \text{//} \\
 & x_1, x_2 & \geq & 0 & \text{//}
\end{array}
$$

Graphische Lösung



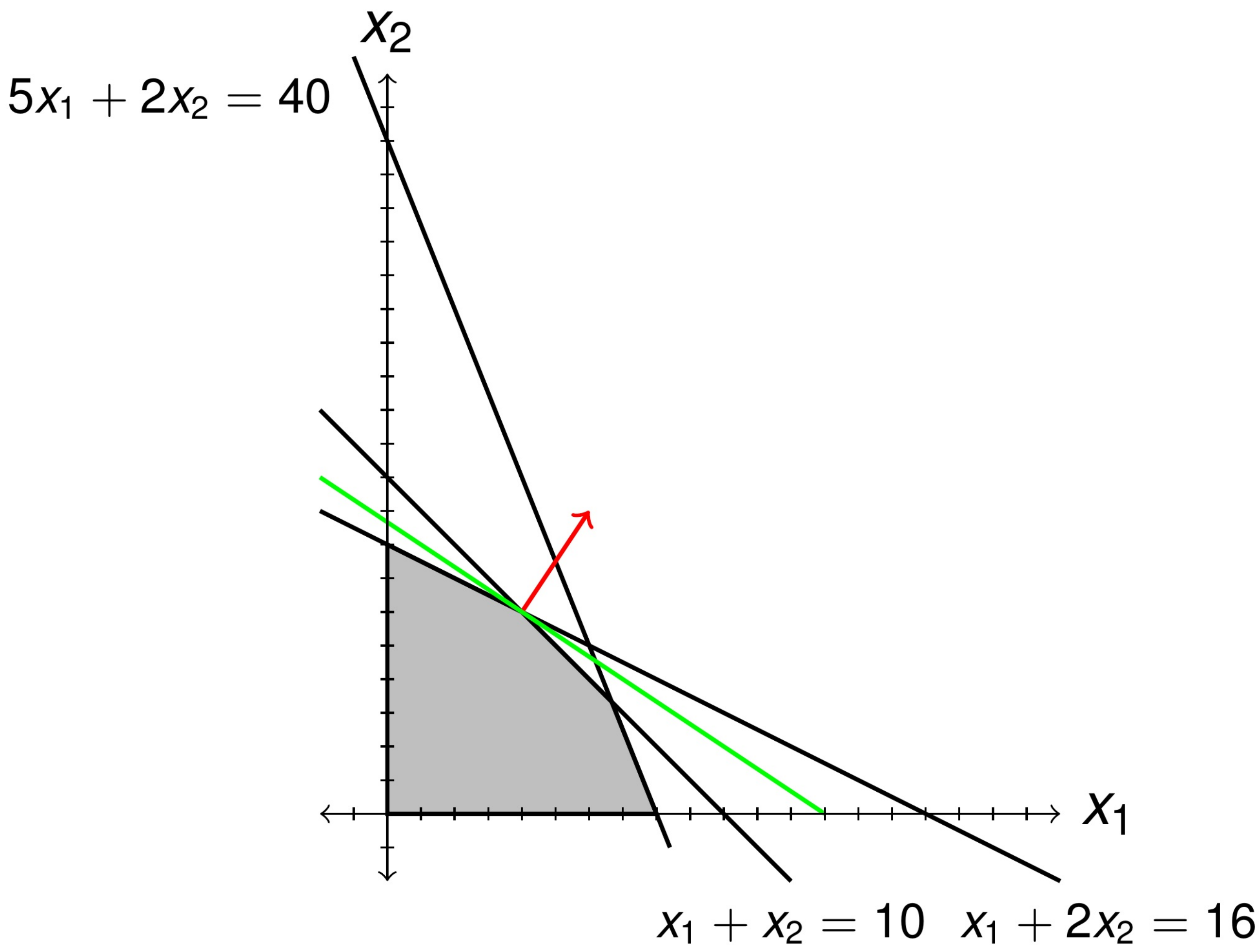
Zulässiger Bereich (graues Gebiet)

Graphische Lösung



Zielfunktion (in rot)

Graphische Lösung



Optimallösung

Definition

- Ein **Optimierungsproblem** ist ein Paar (I, f) aus einer Menge I und $f : I \to \mathbb{R}$.
- Die Elemente von I heißen **zulässige Lösungen** von (I, f).
- Falls $I = \emptyset$, heißt (I, f) **unzulässig**, sonst **zulässig**.
- Die Funktion f heißt **Zielfunktion** von (I, f).
- Wenn (I, f) ein **Maximierungsproblem** ist, suchen wir ein $x^* \in I$ (eine **Optimallösung**), sodass für alle $x \in I$ gilt $f(x) \leq f(x^*)$.
- Das Maximierungsproblem (I, f) heißt **beschränkt**, wenn es ein $K \in \mathbb{R}$ gibt, sodass $f(x) \leq K$ für alle $x \in I$ gilt. Sonst heißt (I, f) **unbeschränkt**.
- Analoge Definitionen für **Minimierungsprobleme**.

LINEARE PROGRAMIERUNG

Eingabe: Eine Matrix $A \in \mathbb{R}^{m \times n}$, Vektoren $c \in \mathbb{R}^n$ und $b \in \mathbb{R}^m$.

Gesucht: Ein Vektor $x \in \mathbb{R}^n$ mit $Ax \leq b$, der $c^t x$ maximiert.

Beispiel:

$$\max \quad (3, -2, 5) \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

$$\text{s.d.} \quad \begin{pmatrix} -2 & 3 & 1 \\ 1 & 1 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \leq \begin{pmatrix} 5 \\ 2 \end{pmatrix}$$

Also (mit der vorigen Definition)

- Zielfunktion $f : \mathbb{R}^n \to \mathbb{R}$ mit $f(x) = c^t x$.
- $I = \{x \in \mathbb{R}^n \mid Ax \leq b\}$.

Varianten

- Minimierungsprobleme können ebenfalls modelliert werden, da $\min\{c^t x \mid Ax \leq b\} = \max\{-c^t x \mid Ax \leq b\}$
- Lineare Gleichungen $a^t x = \beta$ können durch die beiden Ungleichungen $a^t x \leq \beta$ und $a^t x \geq \beta$ modelliert werden.

Aber: Strikte Ungleichungen der Form $a^t x < \beta$ sind verboten.

Notation: Wenn nicht anders gesagt, sei immer
$A = (a_{ij})_{\substack{i=1,\ldots,m \\ j=1,\ldots,n}} \in \mathbb{R}^{m\times n}$, $b = (b_1, \ldots, b_m) \in \mathbb{R}^m$ and
$c = (c_1, \ldots, c_n) \in \mathbb{R}^n$.

Ebenso: $x = (x_1, \ldots, x_n) \in \mathbb{R}^n$

Also: In $\max\{c^t x \mid Ax \leq b\}$ gibt es n Variablen $(x_1, \ldots, x_n)$ und m Nebenbedingungen $\sum_{j=1}^n a_{ij}x_j \leq b_i$ (mit $i = 1, \ldots, m\}$).

Standardformen

Standard-Ungleichungsform:

$$\begin{array}{rrcl} & \max c^t x & & \\ \text{s.t.} & Ax & \leq & b \end{array} \tag{1}$$

Standard-Gleichungsform:

$$\begin{array}{rrcl} & \max c^t x & & \\ \text{s.t.} & Ax & = & b \\ & x & \geq & 0 \end{array} \tag{2}$$

Beide Formen können ineinander transformiert werden.

Von Standard-Gleichungsform in Standard-Ungleichungsform: Ersetze $Ax = b$ durch $Ax \leq b$ und $-Ax \leq -b$, und ersetze $x \geq 0$ durch $-I_n x \leq 0$ (dabei sei I_n die $n \times n$-Einheitsmatrix).

Von Standard-Ungleichungsform in Standard-Gleichungsform

Betrachte LP in Standard-Ungleichungsform

$$\begin{aligned} & \max c^t x \\ \text{s.t.} \quad & Ax \leq b \end{aligned} \tag{3}$$

- Ersetze jede Variable x_i durch zwei Variablen z_i und $\bar{z}_i$.
- Für jede der m Nebenbedingungen fügen wir eine neue Variable $\tilde{x}_i$ ein (eine Schlupfvariable (englisch: slack variable)).

Mit $z = (z_1, \ldots, z_n)$, $\bar{z} = (\bar{z}_1, \ldots, \bar{z}_n)$ and $\tilde{x} = (\tilde{x}_1, \ldots, \tilde{x}_m)$, betrachte:

$$\begin{aligned} & \max c^t (z - \bar{z}) \\ Az - A\bar{z} + \tilde{x} = b \quad \text{s.t.} \quad & [A \,|\, -A \,|\, I_m] \begin{pmatrix} z \\ \bar{z} \\ \tilde{x} \end{pmatrix} = b \\ & z, \bar{z}, \tilde{x} \geq 0 \end{aligned}$$

Beachte: $[A \,|\, -A \,|\, I_m]$ ist die $m \times 2n + m$-Matrix, die aus den Teilmatrizen A, $-A$ and I_m besteht.

Von Standard-Ungleichungsform in Standard-Gleichungsform

$$\begin{array}{rrcl} & \max c^t x & & \\ \text{s.t.} & Ax & \leq & b \end{array} \tag{4}$$

$$\begin{array}{rrcl} & \max c^t(z - \bar{z}) & & \\ \text{s.t.} & [A\,|\,-A\,|\,I_m]\begin{pmatrix} z \\ \bar{z} \\ \tilde{x} \end{pmatrix} & = & b \\ & z, \bar{z}, \tilde{x} & \geq & 0 \end{array} \tag{5}$$

Äquivalenz der LPs:

Jeder Lösung z,$\bar{z}$ und $\tilde{x}$ von (5) gibt eine Lösung von (4) mit denselben Kosten durch $x_j := z_j - \bar{z}_j$ (für $j \in \{1, \ldots, n\}$).

Und: Wenn x eine Lösung von (4) ist, erhalte Lösung von (5) mit denselben Kosten durch $z_j := \max\{x_j, 0\}$, $\bar{z}_j := -\min\{x_j, 0\}$ (für $j \in \{1, \ldots, n\}$) und $\tilde{x}_i = b_i - \sum_{j=1}^n a_{ij}x_j$ (für $i \in \{1, \ldots, m\}$, wobei wieder $\sum_{j=1}^n a_{ij}x_j \leq b_i$ die i-te Nebenbedingung von $Ax \leq b$ ist).

Von Standard-Ungleichungsform in Standard-Gleichungsform

$$\begin{array}{lrcl} & \max c^t x & & \\ \text{s.t.} & Ax & \leq & b \end{array}$$

$$\begin{array}{lrcl} & \max c^t(z - \bar{z}) & & \\ \text{s.t.} & [A \mid -A \mid I_m] \begin{pmatrix} z \\ \bar{z} \\ \tilde{x} \end{pmatrix} & = & b \\ & z, \bar{z}, \tilde{x} & \geq & 0 \end{array}$$

Beachte: Diese Transformation verändert die Zahl der Variablen.

$\Rightarrow$ Die geometrische Struktur des Lösungsraums ändert sich.

Modellierung von Optimierungsproblemen als LPs

Defintion

Sei G ein gerichteter Graph mit Kapazitäten $u : E(G) \to \mathbb{R}_{>0}$ und seien s und t Knoten von G. Ein zulässiger **s-t-Fluss** in (G, u) ist eine Abbildung $f : E(G) \to \mathbb{R}_{\geq 0}$ mit

- $f(e) \leq u(e)$ für alle $e \in E(G)$ und
- $\Delta_f(v) := \sum_{e \in \delta^+_G(v)} f(e) - \sum_{e \in \delta^-_G(v)} f(e) = 0$ für alle $v \in V(G) \setminus \{s, t\}$.

Der **Wert** eines s-t-Flusses f ist $\mathrm{val}(f) = \Delta_f(s)$.

Modellierung von Optimierungsproblemen als LPs

MAXIMUM-FLOW-PROBLEM

Eingabe: Ein gerichteter Graph G, Kapazitäten $u : E(G) \to \mathbb{R}_{>0}$, Knoten $s, t \in V(G)$ mit $s \neq t$.

Aufgabe: Finde einen s-t-Fluss $f : E(G) \to \mathbb{R}_{\geq 0}$ mit maximalem Wert.

LP-Formulierung:

$$
\begin{array}{llcll}
\max & \displaystyle\sum_{e \in \delta^+_G(s)} x_e - \sum_{e \in \delta^-_G(s)} x_e & & & \\
\text{s.t.} & x_e & \geq & 0 & \text{für } e \in E(G) \\
& x_e & \leq & u(e) & \text{für } e \in E(G) \\
& \displaystyle\sum_{e \in \delta^+_G(v)} x_e - \sum_{e \in \delta^-_G(v)} x_e & = & 0 & \text{für } v \in V(G) \setminus \{s, t\}
\end{array}
$$

Modellierung von Optimierungsproblemen als LPs

BOTTLENECK-MAXIMUM-FLOW-PROBLEM MIT 2 QUELLEN

Eingabe: Ein gerichteter Graph G, Kapazitäten $u : E(G) \to \mathbb{R}_{>0}$, drei Knoten $s_1, s_2, t \in V(G)$.

Task: Finde eine Abbildung $f : E(G) \to \mathbb{R}_{\geq 0}$ mit

- $f(e) \leq u(e)$ für alle $e \in E(G)$ und
- $\Delta_f(e) = 0$ für alle $v \in V(G) \setminus \{s_1, s_2, t\}$

sodass $\min\{\Delta_f(s_1), \Delta_f(s_2)\}$ maximiert wird.

Wie kann dieses Problem als LP modelliert werden?

$$\max \quad \min\{c^t x + d,\ e^t x + f\}$$

$$\text{s.d.} \quad Ax \leq b$$

Modellierung als LP:

$$\max \quad \sigma$$

$$\begin{aligned} \text{s.d.} \quad \sigma &\leq c^t x + d \\ \sigma &\leq e^t x + f \\ Ax &\leq b \end{aligned}$$

$$\min \; |c^t x + d|$$

$$\text{s.t.} \quad Ax \leq b$$

$$\max \quad -\sigma$$

$$\text{s.t.} \quad -\sigma - c^t x \leq d$$

$$-\sigma + c^t x \leq -d$$

$$Ax \leq b$$

Ganzzahlige Lineare Programmierung

Ganzzahlige LINEARE PROGRAMMIERUNG

Eingabe: Eine Matrix $A \in \mathbb{R}^{m \times n}$, Vektoren $c \in \mathbb{R}^n$ und $b \in \mathbb{R}^m$.

Aufgabe: Finde einen Vektor $x \in \mathbb{Z}^n$ mit $Ax \leq b$, der $c^t x$ maximiert.

Ganzzahlige Lineare Programme bezeichnen wir als ILPs (Integer Lineare Programs).

Oft müssen nur einige Variablen ganzzahlig sein $\Rightarrow$ GEMISCHT-GANZZAHLIGE PROGRAMMIERUNG (englisch MIXED INTEGER LINEAR PROGRAMMING (MILP))

Wir werden sehen: Ganzzahligkeitsnebenbedingungen machen das Problem (meist) viel schwerer.

Beispiel

VERTEX-COVER-PROBLEM

Eingabe: Ein ungerichteter Graph G, Gewichte $c : V(G) \to \mathbb{R}_{\geq 0}$.

Gesucht: Eine Menge $X \subseteq V(G)$ mit $\{v, w\} \cap X \neq \emptyset$ für alle $e = \{v, w\} \in E(G)$, sodass $\sum_{v \in X} c(v)$ minimiert wird.

Problem ist NP-schwer.
Formulierung als ILP:

$$
\begin{array}{lll}
 & \min \sum_{v \in V(G)} x_v c(v) & \\
s.t. & x_v + x_w \geq 1 & \text{für } \{v, w\} \in E(G) \\
 & x_v \in \{0, 1\} & \text{für } v \in V(G)
\end{array}
\qquad (6)
$$

Wenn $(x_v)_{v \in V(G)}$ eine Optimallösung von ~~(20)~~ (6) ist, dann ist $X = \{v \in V(G) \mid x_v = 1\}$ eine Optimallösung des VERTEX-COVER-PROBLEMS.
$\Rightarrow$ GANZZAHLIGE LINEARE PROGRAMMIERUNG ist selbst NP-schwer.

Beispiel

$$\begin{array}{lll} & \min \sum_{v \in V(G)} x_v c(v) & \\ s.t. & x_v + x_w \geq 1 & \text{für } \{v, w\} \in E(G) \\ & x_v \in \{0, 1\} & \text{für } v \in V(G) \end{array}$$

Idee: Ignoriere die Ganzahligkeitbedingungen ($x_v \in \{0, 1\}$):

$$\begin{array}{lll} & \min \sum_{v \in V(G)} x_v c(v) & \\ s.t. & x_v + x_w \geq 1 & \text{for } \{v, w\} \in E(G) \\ & x_v \geq 0 & \text{for } v \in V(G) \\ & x_v \leq 1 & \text{for } v \in V(G) \end{array} \tag{7}$$

Diese LP heißt **LP-Relaxierung** des ILPs.
Es liefert eine 2-Approximation für VERTEX COVER: Für jede Lösung x des relaxierten Problems, erhalte ganzzahlige Lösung $\tilde{x}$ durch

$$\tilde{x}_v = \begin{cases} 1 & : \quad x_v \geq \frac{1}{2} \\ 0 & : \quad x_v < \frac{1}{2} \end{cases}$$

$\Rightarrow$ Ergibt zulässige ILP-Lösung mit $\sum_{v \in V(G)} \tilde{x}_v c(v) \leq 2 \sum_{v \in V(G)} x_v c(v)$.

Ganzzahlige Lineare Programmierung

- Bei Minimierungsproblemen, nennt man den Quotienten aus optimalem Lösungswert von ILP und seiner LP-Relaxierung Ganzahligkeitslücke (Integrality Gap).
- Bei VERTEX COVER ist er also (höchstens) 2.
- Es gibt ILPs mit beliebig großer Ganzzahligkeitslücke.

Beispiel:

> STABLE-SET-PROBLEM
>
> *Eingabe:* Ein ungerichteter Graph G, Gewichte $c : V(G) \to \mathbb{R}_{\geq 0}$.
>
> *Gesucht:* Eine Menge $X \subseteq V(G)$ mit $|\{v, w\} \cap X| \leq 1$ für alle $e = \{v, w\} \in E(G)$, sodass $\sum_{v \in X} c(v)$ maximiert wird.

ILP-Formulierung:

$$
\begin{array}{lll}
 & \max \sum_{v \in V(G)} x_v c(v) & \\
s.t. & x_v + x_w \leq 1 & \text{für } \{v, w\} \in E(G) \\
 & x_v \in \{0, 1\} & \text{für } v \in V(G)
\end{array}
\tag{8}
$$

LP-Relaxierung:

$$
\begin{array}{lll}
 & \max \sum_{v \in V(G)} x_v c(v) & \\
s.t. & x_v + x_w \leq 1 & \text{for } \{v, w\} \in E(G) \\
 & x_v \geq 0 & \text{for } v \in V(G) \\
 & x_v \leq 1 & \text{for } v \in V(G)
\end{array}
\tag{9}
$$

Beispiel:

ILP-Formulierung:

$$\begin{array}{lll} & \max \sum_{v \in V(G)} x_v c(v) & \\ s.t. & x_v + x_w \leq 1 & \text{für } \{v, w\} \in E(G) \\ & x_v \in \{0, 1\} & \text{für } v \in V(G) \end{array} \tag{10}$$

LP-Relaxierung:

$$\begin{array}{lll} & \max \sum_{v \in V(G)} x_v c(v) & \\ s.t. & x_v + x_w \leq 1 & \text{for } \{v, w\} \in E(G) \\ & x_v \geq 0 & \text{for } v \in V(G) \\ & x_v \leq 1 & \text{for } v \in V(G) \end{array} \tag{11}$$

Wenn G ein vollständiger Graph ist und $c(v) = 1$ für alle $v \in V(G)$ gilt, dann ergibt $x_v = \frac{1}{2}$ (für alle $v \in V(G)$) eine LP-Lösung mit Wert $\frac{n}{2}$. Optimaler ganzzahliger Lösungswert ist aber 1.

$\Rightarrow$ Ganzzahligkeitslücke ist (mindestens) $\frac{n}{2}$.

Polyeder

Definition

Sei $X \subseteq \mathbb{R}^n$ (für $n \in \mathbb{N}$). X heißt **konvex**, wenn für alle $x, y \in X$ und $t \in [0, 1]$ gilt: $tx + (1-t)y \in X$.

Definition

Für $x_1, \ldots, x_k \in \mathbb{R}^n$, $\lambda_1, \ldots, \lambda_k$, $\lambda_i \geq 0$ $(i \in \{1, \ldots, k\})$ mit $\sum_{i=1}^k \lambda_i = 1$ heißt $x = \sum_{i=1}^k \lambda_i x_i$ **Konvexkombination** von $x_1, \ldots, x_k$. Die **konvexe Hülle** $\mathrm{conv}(X)$ eine Menge $X \subseteq \mathbb{R}^n$ ist die Menge aller Konvexkombinationen von Mengen von Vektoren in X.

Man sieht leicht: Die konvexe Hülle einer Menge $X \subseteq \mathbb{R}^n$ ist die (inklusionsweise) kleinste konvexe Menge, die X enthält.