## 6 Lineare Programmierung

# 6 Lineare Programmierung

**Lineares Programm (LP)**: Finde optimale Werte für $d$ **reelle Variablen** $x_1, \ldots, x_d \in \mathbb{R}$.
Dabei soll eine **lineare Zielfunktion**

$$c_1 x_1 + \ldots + c_d x_d$$

für gegebene Koeffizienten $c_1, \ldots, c_d \in \mathbb{R}$ **minimiert** oder **maximiert** werden.

**Lineares Programm (LP)**: Finde optimale Werte für $d$ **reelle Variablen** $x_1, \ldots, x_d \in \mathbb{R}$. Dabei soll eine **lineare Zielfunktion**

$$c_1 x_1 + \ldots + c_d x_d$$

für gegebene Koeffizienten $c_1, \ldots, c_d \in \mathbb{R}$ **minimiert** oder **maximiert** werden.

Es müssen $m$ **lineare Nebenbedingungen** eingehalten werden. Für jedes $i \in \{1, \ldots, m\}$ sind Koeffizienten $a_{i1}, \ldots, a_{id} \in \mathbb{R}$ und $b_i \in \mathbb{R}$ gegeben. Eine Belegung der Variablen ist nur dann gültig, wenn sie die folgenden Nebenbedingungen einhält:

$$\begin{aligned} a_{11} x_1 + \ldots + a_{1d} x_d &\leq b_1 \\ &\vdots \\ a_{m1} x_1 + \ldots + a_{md} x_d &\leq b_m \end{aligned}$$

**Lineares Programm (LP)**: Finde optimale Werte für $d$ **reelle Variablen** $x_1, \ldots, x_d \in \mathbb{R}$. Dabei soll eine **lineare Zielfunktion**

$$c_1 x_1 + \ldots + c_d x_d$$

für gegebene Koeffizienten $c_1, \ldots, c_d \in \mathbb{R}$ **minimiert** oder **maximiert** werden.

Es müssen $m$ **lineare Nebenbedingungen** eingehalten werden. Für jedes $i \in \{1, \ldots, m\}$ sind Koeffizienten $a_{i1}, \ldots, a_{id} \in \mathbb{R}$ und $b_i \in \mathbb{R}$ gegeben. Eine Belegung der Variablen ist nur dann gültig, wenn sie die folgenden Nebenbedingungen einhält:

$$\begin{aligned} a_{11} x_1 + \ldots + a_{1d} x_d &\leq b_1 \\ &\vdots \\ a_{m1} x_1 + \ldots + a_{md} x_d &\leq b_m \end{aligned}$$

Statt $\leq$ ist auch $\geq$ erlaubt.

Sei $x^T = (x_1, \ldots, x_d)$ und $c^T = (c_1, \ldots, c_d)$.

Damit kann die **Zielfunktion als Skalarprodukt** $c \cdot x$ geschrieben werden.

Sei $x^\mathsf{T} = (x_1, \ldots, x_d)$ und $c^\mathsf{T} = (c_1, \ldots, c_d)$.
Damit kann die **Zielfunktion als Skalarprodukt** $c \cdot x$ geschrieben werden.

Außerdem sei $A \in \mathbb{R}^{m \times d}$ die Matrix mit den Einträgen $a_{ij}$ und $b^\mathsf{T} = (b_1, \ldots, b_m) \in \mathbb{R}^m$.
Dann **entspricht jede Zeile der Matrix einer Nebenbedingung**.

Wir können die Nebenbedingungen als $Ax \leq b$ schreiben.

Sei $x^T = (x_1, \ldots, x_d)$ und $c^T = (c_1, \ldots, c_d)$.
Damit kann die **Zielfunktion als Skalarprodukt** $c \cdot x$ geschrieben werden.

Außerdem sei $A \in \mathbb{R}^{m \times d}$ die Matrix mit den Einträgen $a_{ij}$ und $b^T = (b_1, \ldots, b_m) \in \mathbb{R}^m$.
Dann **entspricht jede Zeile der Matrix einer Nebenbedingung**.

Wir können die Nebenbedingungen als $Ax \leq b$ schreiben.

**Lineares Programm**

| | |
|---|---|
| **Eingabe:** | $c \in \mathbb{R}^d$, $b \in \mathbb{R}^m$, $A \in \mathbb{R}^{m \times d}$ |
| **Lösungen:** | alle $x \in \mathbb{R}^d$ mit $Ax \leq b$ |
| **Zielfunktion:** | minimiere/maximiere $c \cdot x$ |

**Beispiel:**

Ein Hobbygärtner besitzt **100 m$^2$ Land**, auf dem er **Blumen** und **Gemüse** anbauen möchte.

- 60 m$^2$ Land **geeignet für Blumen und Gemüse**
  40 m$^2$ **nur für Gemüse geeignet**
- Er hat ein **Budget** in Höhe von 720 €.
- **Kosten für Blumen:** 9 € pro Quadratmeter
  **Kosten für Gemüse:** 6 € pro Quadratmeter
- **Erlös für Blumen:** 20 € pro Quadratmeter
  **Erlös für Gemüse:** 10 € pro Quadratmeter

**Frage:** Wie viele Blumen und wie viel Gemüse sollte er anbauen, um seinen **Gewinn zu maximieren**?

**Wähle geeignete Variablen:**

- $x_B \in \mathbb{R} =$ **Quadratmeter Blumen**
- $x_G \in \mathbb{R} =$ **Quadratmeter Gemüse**

Dann können wir die Nebenbedingungen wie folgt formulieren:

| | |
|---|---|
| $x_B \geq 0, x_G \geq 0$ | (keine negative Anbaufläche) |
| $x_B + x_G \leq 100$ | (maximal 100 $m^2$) |
| $x_B \leq 60$ | (maximal 60 $m^2$ Blumen) |
| $9x_B + 6x_G \leq 720$ | (Budget von 720 €) |

**Wähle geeignete Variablen:**

- $x_B \in \mathbb{R} =$ **Quadratmeter Blumen**
- $x_G \in \mathbb{R} =$ **Quadratmeter Gemüse**

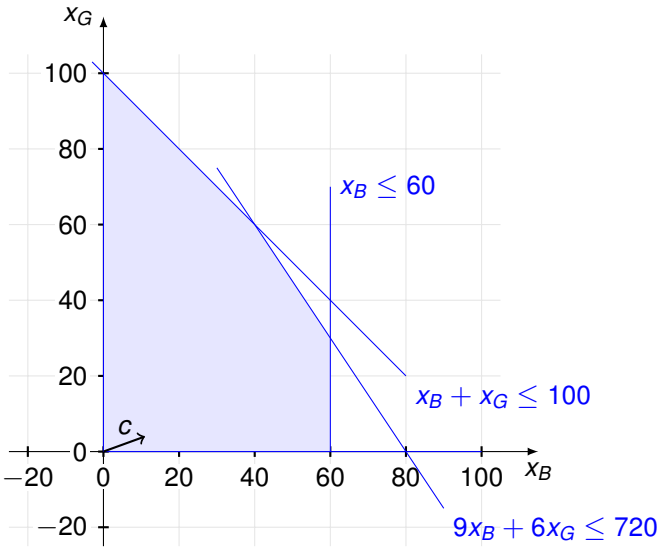
Dann können wir die Nebenbedingungen wie folgt formulieren:

| | |
|---|---|
| $x_B \geq 0, x_G \geq 0$ | (keine negative Anbaufläche) |
| $x_B + x_G \leq 100$ | (maximal 100 $m^2$) |
| $x_B \leq 60$ | (maximal 60 $m^2$ Blumen) |
| $9x_B + 6x_G \leq 720$ | (Budget von 720 €) |

Der Gewinn ist die **Differenz von Erlös und Ausgaben**.

Somit soll die folgende Zielfunktion maximiert werden:

$$(20 - 9)x_B + (10 - 6)x_G = 11x_B + 4x_G.$$


$x_G$
100
80
60
40
20
0
−20
−20
0
20
40
60
80
100
$x_B$
$x_B \leq 60$
$x_B + x_G \leq 100$
$9x_B + 6x_G \leq 720$
$c$

**Beispiel: Maximaler Fluss**

**Eingabe:** Flussnetzwerk $G = (V, E)$ mit Quelle $s \in V$ und Senke $t \in V$, Kapazitätsfunktion $c : E \to \mathbb{N}_0$

**Aufgabe:** Finde einen maximalen Fluss von $s$ nach $t$ in $G$.

**Beispiel: Maximaler Fluss**

**Eingabe:** Flussnetzwerk $G = (V, E)$ mit Quelle $s \in V$ und Senke $t \in V$, Kapazitätsfunktion $c : E \to \mathbb{N}_0$

**Aufgabe:** Finde einen maximalen Fluss von $s$ nach $t$ in $G$.

**Modellierung als LP:**

**Variablen:** Für jedes $e \in E$ Variable $x_e \in \mathbb{R}$, die den Fluss auf $e$ angibt.

**Beispiel: Maximaler Fluss**

**Eingabe:** Flussnetzwerk $G = (V, E)$ mit Quelle $s \in V$ und Senke $t \in V$, Kapazitätsfunktion $c : E \to \mathbb{N}_0$

**Aufgabe:** Finde einen maximalen Fluss von $s$ nach $t$ in $G$.

**Modellierung als LP:**

**Variablen:** Für jedes $e \in E$ Variable $x_e \in \mathbb{R}$, die den Fluss auf $e$ angibt.

**Zielfunktion:**

$$\sum_{e=(s,v)} x_e - \sum_{e=(v,s)} x_e$$

**Beispiel: Maximaler Fluss**

**Eingabe:** Flussnetzwerk $G = (V, E)$ mit Quelle $s \in V$ und Senke $t \in V$, Kapazitätsfunktion $c : E \to \mathbb{N}_0$

**Aufgabe:** Finde einen maximalen Fluss von $s$ nach $t$ in $G$.

**Modellierung als LP:**

**Variablen:** Für jedes $e \in E$ Variable $x_e \in \mathbb{R}$, die den Fluss auf $e$ angibt.

**Zielfunktion:**

$$\sum_{e=(s,v)} x_e - \sum_{e=(v,s)} x_e$$

**Nebenbedingungen:**

$\forall e \in E : x_e \geq 0$ (Fluss nicht negativ)

$\forall e \in E : x_e \leq c(e)$ (Fluss nicht größer als Kapazität)

$\forall v \in V \setminus \{s, t\} : \sum_{e=(u,v)} x_e - \sum_{e=(v,u)} x_e = 0$ (Flusserhaltung)

# 6.1 Grundlagen

**kanonische Form**

$$\min c \cdot x$$

$$Ax \leq b$$

$$x \geq 0$$

**Gleichungsform**

$$\min c \cdot x$$

$$Ax = b$$

$$x \geq 0$$

**kanonische Form**

$$\min c \cdot x$$
$$Ax \leq b$$
$$x \geq 0$$

**Gleichungsform**

$$\min c \cdot x$$
$$Ax = b$$
$$x \geq 0$$

Sei $a_i = (a_{i1}, \ldots, a_{id})^T \in \mathbb{R}^d$ die $i$-te Zeile von $A$.

# 6.1 Grundlagen

| kanonische Form | Gleichungsform |
| --- | --- |
| $\min c \cdot x$ | $\min c \cdot x$ |
| $Ax \leq b$ | $Ax = b$ |
| $x \geq 0$ | $x \geq 0$ |

Sei $a_i = (a_{i1}, \ldots, a_{id})^{\mathsf{T}} \in \mathbb{R}^d$ die $i$-te Zeile von $A$.

## Transformationen

- „maximiere $c \cdot x$“ entspricht „minimiere $-c \cdot x$“.

# 6.1 Grundlagen

| kanonische Form | Gleichungsform |
| --- | --- |
| $\min c \cdot x$ | $\min c \cdot x$ |
| $Ax \leq b$ | $Ax = b$ |
| $x \geq 0$ | $x \geq 0$ |

Sei $a_i = (a_{i1}, \ldots, a_{id})^\mathsf{T} \in \mathbb{R}^d$ die $i$-te Zeile von $A$.

## Transformationen

- „maximiere $c \cdot x$“ entspricht „minimiere $-c \cdot x$“.
- Variable $x_i$ kann durch $x_i' - x_i''$ für zwei Variablen $x_i' \geq 0$ und $x_i'' \geq 0$ ersetzt werden.

# 6.1 Grundlagen

| kanonische Form | Gleichungsform |
|---|---|
| $\min c \cdot x$ | $\min c \cdot x$ |
| $Ax \leq b$ | $Ax = b$ |
| $x \geq 0$ | $x \geq 0$ |

Sei $a_i = (a_{i1}, \ldots, a_{id})^{\mathsf{T}} \in \mathbb{R}^d$ die $i$-te Zeile von $A$.

## Transformationen

- „maximiere $c \cdot x$“ entspricht „minimiere $-c \cdot x$“.
- Variable $x_i$ kann durch $x_i' - x_i''$ für zwei Variablen $x_i' \geq 0$ und $x_i'' \geq 0$ ersetzt werden.
- „$a_i \cdot x \geq b_i$“ entspricht „$-a_i \cdot x \leq -b_i$“.

# 6.1 Grundlagen

| kanonische Form | Gleichungsform |
| --- | --- |
| $\min c \cdot x$ | $\min c \cdot x$ |
| $Ax \leq b$ | $Ax = b$ |
| $x \geq 0$ | $x \geq 0$ |

Sei $a_i = (a_{i1}, \ldots, a_{id})^T \in \mathbb{R}^d$ die $i$-te Zeile von $A$.

## Transformationen

- „maximiere $c \cdot x$“ entspricht „minimiere $-c \cdot x$“.
- Variable $x_i$ kann durch $x_i' - x_i''$ für zwei Variablen $x_i' \geq 0$ und $x_i'' \geq 0$ ersetzt werden.
- „$a_i \cdot x \geq b_i$“ entspricht „$-a_i \cdot x \leq -b_i$“.
- Gleichung $a_i \cdot x = b_i$ kann durch $a_i \cdot x \leq b_i$ und $a_i \cdot x \geq b_i$ ersetzt werden.

| kanonische Form | Gleichungsform |
|---|---|
| $\min c \cdot x$ | $\min c \cdot x$ |
| $Ax \leq b$ | $Ax = b$ |
| $x \geq 0$ | $x \geq 0$ |

Sei $a_i = (a_{i1}, \ldots, a_{id})^T \in \mathbb{R}^d$ die $i$-te Zeile von $A$.

**Transformationen**

- „maximiere $c \cdot x$“ entspricht „minimiere $-c \cdot x$“.
- Variable $x_i$ kann durch $x_i' - x_i''$ für zwei Variablen $x_i' \geq 0$ und $x_i'' \geq 0$ ersetzt werden.
- „$a_i \cdot x \geq b_i$“ entspricht „$-a_i \cdot x \leq -b_i$“.
- Gleichung $a_i \cdot x = b_i$ kann durch $a_i \cdot x \leq b_i$ und $a_i \cdot x \geq b_i$ ersetzt werden.
- $a_i \cdot x \leq b_i$ können wir durch $s_i + a_i \cdot x = b_i$ für eine **Schlupfvariable** $s_i \geq 0$ darstellen.

**Geometrische Interpretation:** Betrachte LP in kanonischer Form

Variablenbelegung $x \in \mathbb{R}^d$ **entspricht Punkt im** $\mathbb{R}^d$.

**Geometrische Interpretation:** Betrachte LP in kanonischer Form

Variablenbelegung $x \in \mathbb{R}^d$ **entspricht Punkt im** $\mathbb{R}^d$.

Eine Gleichung $a_i \cdot x = b_i$ definiert eine **affine Hyperebene** $\{x \in \mathbb{R}^d \mid a_i \cdot x = b_i\}$.

Jede solche affine Hyperebene definiert den **abgeschlossenen Halbraum**

$$\mathcal{H}_i = \{x \in \mathbb{R}^d \mid a_i \cdot x \leq b_i\}.$$

**Geometrische Interpretation:** Betrachte LP in kanonischer Form

Variablenbelegung $x \in \mathbb{R}^d$ **entspricht Punkt im** $\mathbb{R}^d$.

Eine Gleichung $a_i \cdot x = b_i$ definiert eine **affine Hyperebene** $\{x \in \mathbb{R}^d \mid a_i \cdot x = b_i\}$.

Jede solche affine Hyperebene definiert den **abgeschlossenen Halbraum**

$$\mathcal{H}_i = \{x \in \mathbb{R}^d \mid a_i \cdot x \leq b_i\}.$$

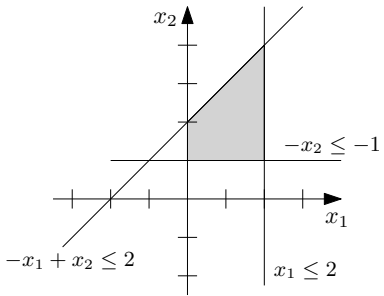
Eine Variablenbelegung $x \in \mathbb{R}^d$ **erfüllt genau dann Nebenbedingung i**, wenn $x \in \mathcal{H}_i$ gilt.

Eine Variablenbelegung $x \in \mathbb{R}^d$ ist genau dann **gültig**, wenn
$x \in \mathcal{P} := \mathcal{H}_1 \cap \ldots \cap \mathcal{H}_m \cap \mathbb{R}^d_{\geq 0}$ gilt.

**Beispiel:**

Betrachte lineares Programm mit den folgenden Nebenbedingungen

$$x_1 \geq 0, \qquad x_2 \geq 0,$$
$$x_1 \leq 2, \quad -x_2 \leq -1,$$
$$-x_1 + x_2 \leq 2$$



Wir können $\mathbb{R}^d_{\geq 0}$ als einen Schnitt von $d$ Halbräumen darstellen:

$$\mathbb{R}^d_{\geq 0} = \{x \in \mathbb{R}^d \mid -x_1 \leq 0\} \cap \ldots \cap \{x \in \mathbb{R}^d \mid -x_d \leq 0\}.$$

Somit ist $\mathcal{P}$ der **Schnitt von endlich vielen Halbräumen**.

Wir können $\mathbb{R}^d_{\geq 0}$ als einen Schnitt von $d$ Halbräumen darstellen:

$$\mathbb{R}^d_{\geq 0} = \{x \in \mathbb{R}^d \mid -x_1 \leq 0\} \cap \ldots \cap \{x \in \mathbb{R}^d \mid -x_d \leq 0\}.$$

Somit ist $\mathcal{P}$ der **Schnitt von endlich vielen Halbräumen**.

Einen solchen Schnitt nennt man **Polyeder**. Wir sagen, dass ein lineares Programm **zulässig** ist, wenn sein **Lösungspolyeder** nichtleer ist.

Wir können $\mathbb{R}^d_{\geq 0}$ als einen Schnitt von $d$ Halbräumen darstellen:

$$\mathbb{R}^d_{\geq 0} = \{x \in \mathbb{R}^d \mid -x_1 \leq 0\} \cap \ldots \cap \{x \in \mathbb{R}^d \mid -x_d \leq 0\}.$$

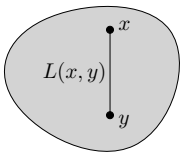
Somit ist $\mathcal{P}$ der **Schnitt von endlich vielen Halbräumen**.

Einen solchen Schnitt nennt man **Polyeder**. Wir sagen, dass ein lineares Programm **zulässig** ist, wenn sein **Lösungspolyeder** nichtleer ist.

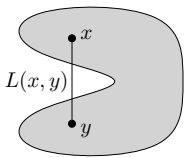
Eine Menge $X$ heißt **konvex**, wenn für alle Punkte $x \in X$ und $y \in X$ gilt:

$$L(x, y) := \{\lambda x + (1 - \lambda)y \mid \lambda \in [0, 1]\} \subseteq X.$$

konvexe Menge




nichtkonvexe Menge



**Lemma 6.1**

Das Lösungspolyeder $\mathcal{P}$ ist konvex.

**Beweis:** Sei $\mathcal{P}$ der Durchschnitt der abgeschlossenen Halbräume $\mathcal{H}_1, \ldots, \mathcal{H}_n$.

**Lemma 6.1**

Das Lösungspolyeder $\mathcal{P}$ ist konvex.

**Beweis:** Sei $\mathcal{P}$ der Durchschnitt der abgeschlossenen Halbräume $\mathcal{H}_1, \ldots, \mathcal{H}_n$.

**Jeder abgeschlossene Halbraum $\mathcal{H} = \{z \mid u \cdot z \leq w\}$ ist konvex:**

Seien $x \in \mathcal{H}$ und $y \in \mathcal{H}$ beliebig. Für jedes $\lambda \in [0, 1]$ gehört dann auch der Punkt $\lambda x + (1 - \lambda)y$ zu $\mathcal{H}$, denn

$$u \cdot (\lambda x + (1 - \lambda)y) = \lambda(u \cdot x) + (1 - \lambda)(u \cdot y) \leq \lambda w + (1 - \lambda)w = w.$$

**Lemma 6.1**

Das Lösungspolyeder $\mathcal{P}$ ist konvex.

**Beweis:** Sei $\mathcal{P}$ der Durchschnitt der abgeschlossenen Halbräume $\mathcal{H}_1, \ldots, \mathcal{H}_n$.

**Jeder abgeschlossene Halbraum $\mathcal{H} = \{z \mid u \cdot z \leq w\}$ ist konvex:**

Seien $x \in \mathcal{H}$ und $y \in \mathcal{H}$ beliebig. Für jedes $\lambda \in [0, 1]$ gehört dann auch der Punkt $\lambda x + (1 - \lambda)y$ zu $\mathcal{H}$, denn

$$u \cdot (\lambda x + (1 - \lambda)y) = \lambda(u \cdot x) + (1 - \lambda)(u \cdot y) \leq \lambda w + (1 - \lambda)w = w.$$

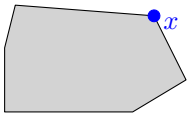
**Der Schnitt zweier konvexer Mengen ist wieder konvex:**

Seien $X \subseteq \mathbb{R}^d$ und $Y \subseteq \mathbb{R}^d$ konvexe Mengen, und seien $x, y \in X \cap Y$ beliebig. Da die Mengen $X$ und $Y$ konvex sind, gilt $L(x, y) \subseteq X$ und $L(x, y) \subseteq Y$. Dementsprechend gilt auch $L(x, y) \subseteq X \cap Y$. □

# 6.1 Grundlagen

Sei ein lineares Programm in kanonischer Form mit Lösungspolyeder $\mathcal{P}$ gegeben.

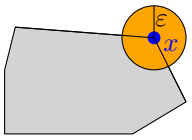
Wir sagen, eine Variablenbelegung $x \in \mathcal{P}$ ist **lokal optimal**, wenn es ein $\varepsilon > 0$ gibt, für das es kein $y \in \mathcal{P}$ mit $\|x - y\| \leq \varepsilon$ und $c \cdot y < c \cdot x$ gibt[1].




[1] Hierbei bezeichnet $\|x - y\|$ den euklidischen Abstand zwischen $x$ und $y$, also $\|x - y\| = \sqrt{(x_1 - y_1)^2 + \ldots + (x_d - y_d)^2}$.

Sei ein lineares Programm in kanonischer Form mit Lösungspolyeder $\mathcal{P}$ gegeben.

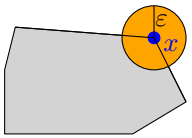
Wir sagen, eine Variablenbelegung $x \in \mathcal{P}$ ist **lokal optimal**, wenn es ein $\varepsilon > 0$ gibt, für das es kein $y \in \mathcal{P}$ mit $\|x - y\| \leq \varepsilon$ und $c \cdot y < c \cdot x$ gibt[1].




[1] Hierbei bezeichnet $\|x - y\|$ den euklidischen Abstand zwischen $x$ und $y$, also $\|x - y\| = \sqrt{(x_1 - y_1)^2 + \ldots + (x_d - y_d)^2}$.

Sei ein lineares Programm in kanonischer Form mit Lösungspolyeder $\mathcal{P}$ gegeben.

Wir sagen, eine Variablenbelegung $x \in \mathcal{P}$ ist **lokal optimal**, wenn es ein $\varepsilon > 0$ gibt, für das es kein $y \in \mathcal{P}$ mit $\|x - y\| \leq \varepsilon$ und $c \cdot y < c \cdot x$ gibt[1].




**Theorem 6.2**

Sei ein lineares Programm in kanonischer Form mit Lösungspolyeder $\mathcal{P}$ gegeben und sei $x \in \mathcal{P}$ eine lokal optimale Variablenbelegung. Dann ist $x$ auch global optimal, d. h. es gibt kein $y \in \mathcal{P}$ mit $c \cdot y < c \cdot x$.

[1] Hierbei bezeichnet $\|x - y\|$ den euklidischen Abstand zwischen $x$ und $y$, also $\|x - y\| = \sqrt{(x_1 - y_1)^2 + \ldots + (x_d - y_d)^2}$.

**Beweis:** Annahme: es gibt $y \in \mathcal{P}$ mit $c \cdot y < c \cdot x$.

**Beweis:** Annahme: es gibt $y \in \mathcal{P}$ mit $c \cdot y < c \cdot x$.

Da $\mathcal{P}$ konvex ist, liegt jeder Punkt aus $L(x, y)$ in $\mathcal{P}$.
Sei $z = \lambda x + (1 - \lambda)y$ mit $\lambda \in [0, 1)$ ein solcher Punkt.

**Beweis:** Annahme: es gibt $y \in \mathcal{P}$ mit $c \cdot y < c \cdot x$.

Da $\mathcal{P}$ konvex ist, liegt jeder Punkt aus $L(x, y)$ in $\mathcal{P}$.
Sei $z = \lambda x + (1 - \lambda)y$ mit $\lambda \in [0, 1)$ ein solcher Punkt.

Zielfunktion an der Stelle $z$:

$$c \cdot z = c \cdot (\lambda x + (1 - \lambda)y) = \lambda(c \cdot x) + (1 - \lambda)(c \cdot y) < c \cdot x.$$

**Beweis:** Annahme: es gibt $y \in \mathcal{P}$ mit $c \cdot y < c \cdot x$.

Da $\mathcal{P}$ konvex ist, liegt jeder Punkt aus $L(x, y)$ in $\mathcal{P}$.
Sei $z = \lambda x + (1 - \lambda)y$ mit $\lambda \in [0, 1)$ ein solcher Punkt.

Zielfunktion an der Stelle $z$:

$$c \cdot z = c \cdot (\lambda x + (1 - \lambda)y) = \lambda(c \cdot x) + (1 - \lambda)(c \cdot y) < c \cdot x.$$

$\Rightarrow$ Jeder Punkt $z \in L(x, y)$ mit $z \neq x$ ist eine echt bessere Variablenbelegung als $x$. $\square$

Ein LP heißt **unbeschränkt**, wenn der zu minimierende Zielfunktionswert innerhalb des Lösungspolyeders $\mathcal{P}$ beliebig klein werden kann. Ansonsten heißt es **beschränkt**.

Ein LP heißt **unbeschränkt**, wenn der zu minimierende Zielfunktionswert innerhalb des Lösungspolyeders $\mathcal{P}$ beliebig klein werden kann. Ansonsten heißt es **beschränkt**.

Sei $c \cdot x$ eine beliebige lineare Zielfunktion und sei $w \in \mathbb{R}$ beliebig. Die Menge

$$\{x \in \mathbb{R}^d \mid c \cdot x = w\}$$

bildet eine **affine Hyperebene mit Normalenvektor c**.

## 6.1 Grundlagen

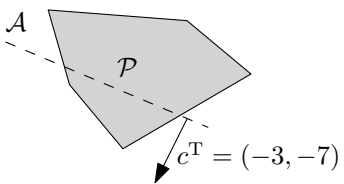
Ein LP heißt **unbeschränkt**, wenn der zu minimierende Zielfunktionswert innerhalb des Lösungspolyeders $\mathcal{P}$ beliebig klein werden kann. Ansonsten heißt es **beschränkt**.

Sei $c \cdot x$ eine beliebige lineare Zielfunktion und sei $w \in \mathbb{R}$ beliebig. Die Menge

$$\{x \in \mathbb{R}^d \mid c \cdot x = w\}$$

bildet eine **affine Hyperebene mit Normalenvektor c**.

1. Finde ein $w \in \mathbb{R}$, sodass $\{x \in \mathbb{R}^d \mid c \cdot x = w\} \cap \mathcal{P} \neq \emptyset$.



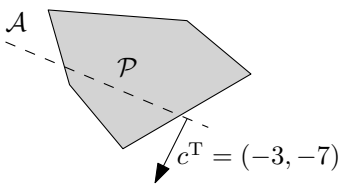
Ein LP heißt **unbeschränkt**, wenn der zu minimierende Zielfunktionswert innerhalb des Lösungspolyeders $\mathcal{P}$ beliebig klein werden kann. Ansonsten heißt es **beschränkt**.

Sei $c \cdot x$ eine beliebige lineare Zielfunktion und sei $w \in \mathbb{R}$ beliebig. Die Menge

$$\{x \in \mathbb{R}^d \mid c \cdot x = w\}$$

bildet eine **affine Hyperebene mit Normalenvektor c**.

1. Finde ein $w \in \mathbb{R}$, sodass $\{x \in \mathbb{R}^d \mid c \cdot x = w\} \cap \mathcal{P} \neq \emptyset$.
2. Verschiebe $\{x \in \mathbb{R}^d \mid c \cdot x = w\}$ solange parallel in Richtung $-c$ wie obiger Schnitt nichtleer ist.



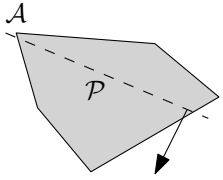
Ein LP heißt **unbeschränkt**, wenn der zu minimierende Zielfunktionswert innerhalb des Lösungspolyeders $\mathcal{P}$ beliebig klein werden kann. Ansonsten heißt es **beschränkt**.

Sei $c \cdot x$ eine beliebige lineare Zielfunktion und sei $w \in \mathbb{R}$ beliebig. Die Menge

$$\{x \in \mathbb{R}^d \mid c \cdot x = w\}$$

bildet eine **affine Hyperebene mit Normalenvektor c**.

1. Finde ein $w \in \mathbb{R}$, sodass $\{x \in \mathbb{R}^d \mid c \cdot x = w\} \cap \mathcal{P} \neq \emptyset$.
2. Verschiebe $\{x \in \mathbb{R}^d \mid c \cdot x = w\}$ solange parallel in Richtung $-c$ wie obiger Schnitt nichtleer ist.



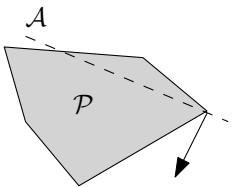
Ein LP heißt **unbeschränkt**, wenn der zu minimierende Zielfunktionswert innerhalb des Lösungspolyeders $\mathcal{P}$ beliebig klein werden kann. Ansonsten heißt es **beschränkt**.

Sei $c \cdot x$ eine beliebige lineare Zielfunktion und sei $w \in \mathbb{R}$ beliebig. Die Menge

$$\{x \in \mathbb{R}^d \mid c \cdot x = w\}$$

bildet eine **affine Hyperebene mit Normalenvektor c**.

1. Finde ein $w \in \mathbb{R}$, sodass $\{x \in \mathbb{R}^d \mid c \cdot x = w\} \cap \mathcal{P} \neq \emptyset$.
2. Verschiebe $\{x \in \mathbb{R}^d \mid c \cdot x = w\}$ solange parallel in Richtung $-c$ wie obiger Schnitt nichtleer ist.



# 6.1 Grundlagen

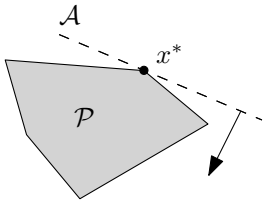
Ein LP heißt **unbeschränkt**, wenn der zu minimierende Zielfunktionswert innerhalb des Lösungspolyeders $\mathcal{P}$ beliebig klein werden kann. Ansonsten heißt es **beschränkt**.

Sei $c \cdot x$ eine beliebige lineare Zielfunktion und sei $w \in \mathbb{R}$ beliebig. Die Menge

$$\{x \in \mathbb{R}^d \mid c \cdot x = w\}$$

bildet eine **affine Hyperebene mit Normalenvektor c**.

1. Finde ein $w \in \mathbb{R}$, sodass $\{x \in \mathbb{R}^d \mid c \cdot x = w\} \cap \mathcal{P} \neq \emptyset$.
2. Verschiebe $\{x \in \mathbb{R}^d \mid c \cdot x = w\}$ solange parallel in Richtung $-c$ wie obiger Schnitt nichtleer ist.



# 6.1 Grundlagen

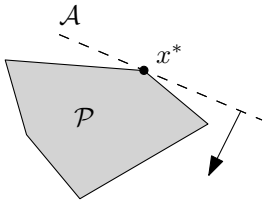
Ein LP heißt **unbeschränkt**, wenn der zu minimierende Zielfunktionswert innerhalb des Lösungspolyeders $\mathcal{P}$ beliebig klein werden kann. Ansonsten heißt es **beschränkt**.

Sei $c \cdot x$ eine beliebige lineare Zielfunktion und sei $w \in \mathbb{R}$ beliebig. Die Menge

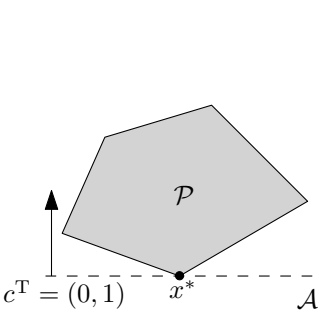
$$\{x \in \mathbb{R}^d \mid c \cdot x = w\}$$

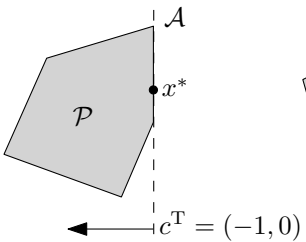
bildet eine **affine Hyperebene mit Normalenvektor c**.

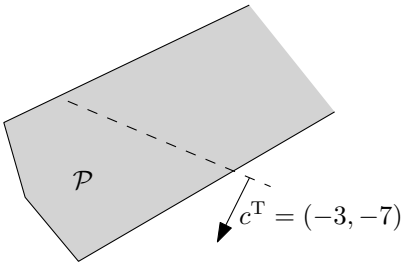
1. Finde ein $w \in \mathbb{R}$, sodass $\{x \in \mathbb{R}^d \mid c \cdot x = w\} \cap \mathcal{P} \neq \emptyset$.
2. Verschiebe $\{x \in \mathbb{R}^d \mid c \cdot x = w\}$ solange parallel in Richtung $-c$ wie obiger Schnitt nichtleer ist.
3. Terminiert der zweite Schritt nicht, so ist das LP unbeschränkt. Ansonsten sei $\mathcal{A} = \{x \in \mathbb{R}^d \mid c \cdot x = w\}$ die letzte Hyperebene mit $\mathcal{A} \cap \mathcal{P} \neq \emptyset$. Dann ist jeder Punkt $x^* \in \mathcal{A} \cap \mathcal{P}$ eine optimale Variablenbelegung des LPs.



**Beispiele:**


$\mathcal{P}$
$c^{\mathrm{T}} = (0, 1)$
$x^*$
$\mathcal{A}$


$\mathcal{A}$
$\mathcal{P}$
$x^*$
$c^{\mathrm{T}} = (-1, 0)$

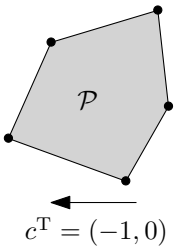

$\mathcal{P}$
$c^{\mathrm{T}} = (-3, -7)$

## Simplex-Algorithmus (informelle Beschreibung)

Starte an einer Ecke $x^1 \in \mathcal{P}$ und teste, ob es eine benachbarte Ecke mit besserem Zielfunktionswert gibt.

Gibt es eine solche benachbarte Ecke $x^2 \in \mathcal{P}$, so mache mit $x^2$ analog weiter und teste, ob es eine **bessere benachbarte Ecke** gibt.

Den Übergang von einer Ecke zu einer besseren in der Nachbarschaft nennen wir auch **Pivotschritt**.

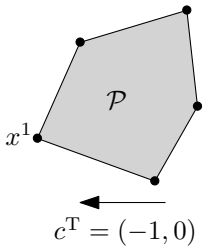


## Simplex-Algorithmus (informelle Beschreibung)

Starte an einer Ecke $x^1 \in \mathcal{P}$ und teste, ob es eine benachbarte Ecke mit besserem Zielfunktionswert gibt.

Gibt es eine solche benachbarte Ecke $x^2 \in \mathcal{P}$, so mache mit $x^2$ analog weiter und teste, ob es eine **bessere benachbarte Ecke** gibt.

Den Übergang von einer Ecke zu einer besseren in der Nachbarschaft nennen wir auch **Pivotschritt**.

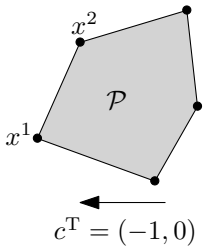


## Simplex-Algorithmus (informelle Beschreibung)

Starte an einer Ecke $x^1 \in \mathcal{P}$ und teste, ob es eine benachbarte Ecke mit besserem Zielfunktionswert gibt.

Gibt es eine solche benachbarte Ecke $x^2 \in \mathcal{P}$, so mache mit $x^2$ analog weiter und teste, ob es eine **bessere benachbarte Ecke** gibt.

Den Übergang von einer Ecke zu einer besseren in der Nachbarschaft nennen wir auch **Pivotschritt**.

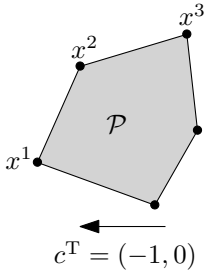


## Simplex-Algorithmus (informelle Beschreibung)

Starte an einer Ecke $x^1 \in \mathcal{P}$ und teste, ob es eine benachbarte Ecke mit besserem Zielfunktionswert gibt.

Gibt es eine solche benachbarte Ecke $x^2 \in \mathcal{P}$, so mache mit $x^2$ analog weiter und teste, ob es eine **bessere benachbarte Ecke** gibt.

Den Übergang von einer Ecke zu einer besseren in der Nachbarschaft nennen wir auch **Pivotschritt**.

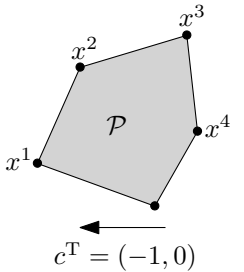


## Simplex-Algorithmus (informelle Beschreibung)

Starte an einer Ecke $x^1 \in \mathcal{P}$ und teste, ob es eine benachbarte Ecke mit besserem Zielfunktionswert gibt.

Gibt es eine solche benachbarte Ecke $x^2 \in \mathcal{P}$, so mache mit $x^2$ analog weiter und teste, ob es eine **bessere benachbarte Ecke** gibt.

Den Übergang von einer Ecke zu einer besseren in der Nachbarschaft nennen wir auch **Pivotschritt**.

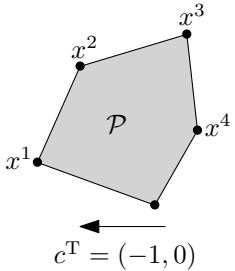


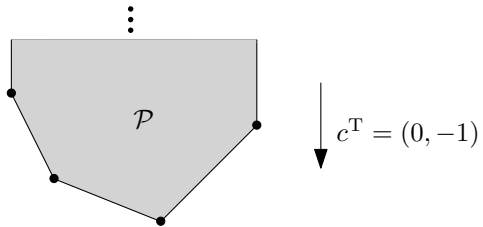
## Simplex-Algorithmus (informelle Beschreibung)

Starte an einer Ecke $x^1 \in \mathcal{P}$ und teste, ob es eine benachbarte Ecke mit besserem Zielfunktionswert gibt.

Gibt es eine solche benachbarte Ecke $x^2 \in \mathcal{P}$, so mache mit $x^2$ analog weiter und teste, ob es eine **bessere benachbarte Ecke** gibt.

Den Übergang von einer Ecke zu einer besseren in der Nachbarschaft nennen wir auch **Pivotschritt**.



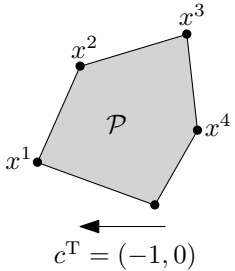



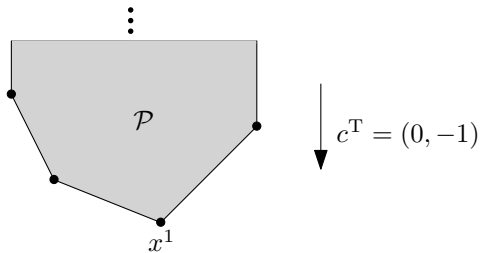
## Simplex-Algorithmus (informelle Beschreibung)

Starte an einer Ecke $x^1 \in \mathcal{P}$ und teste, ob es eine benachbarte Ecke mit besserem Zielfunktionswert gibt.

Gibt es eine solche benachbarte Ecke $x^2 \in \mathcal{P}$, so mache mit $x^2$ analog weiter und teste, ob es eine **bessere benachbarte Ecke** gibt.

Den Übergang von einer Ecke zu einer besseren in der Nachbarschaft nennen wir auch **Pivotschritt**.



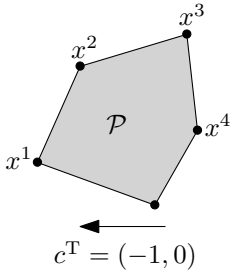



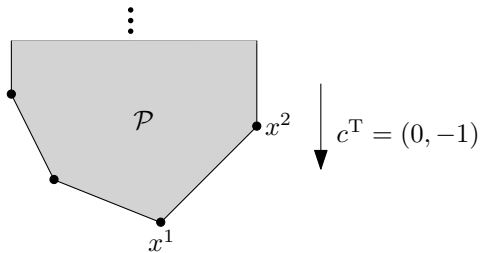
## Simplex-Algorithmus (informelle Beschreibung)

Starte an einer Ecke $x^1 \in \mathcal{P}$ und teste, ob es eine benachbarte Ecke mit besserem Zielfunktionswert gibt.

Gibt es eine solche benachbarte Ecke $x^2 \in \mathcal{P}$, so mache mit $x^2$ analog weiter und teste, ob es eine **bessere benachbarte Ecke** gibt.

Den Übergang von einer Ecke zu einer besseren in der Nachbarschaft nennen wir auch **Pivotschritt**.



### Finden einer initialen Lösung

Sei ein LP mit den Nebenbedingungen $Ax = b$ gegeben und sei o. B. d. A. $b \geq 0$.

### Finden einer initialen Lösung

Sei ein LP mit den Nebenbedingungen $Ax = b$ gegeben und sei o. B. d. A. $b \geq 0$.

Für die $m$ Nebenbedingungen führen wir **Hilfsvariablen** $h_1 \geq 0, \ldots, h_m \geq 0$ ein.
Die NB $a_i \cdot x = b_i$ ersetzen wir für jedes $i$ durch die NB $a_i \cdot x + \mathbf{h_i} = b_i$.
Wir ignorieren die Zielfunktion und definieren als **neue Zielfunktion** $h_1 + \ldots + h_m$.

### Finden einer initialen Lösung

Sei ein LP mit den Nebenbedingungen $Ax = b$ gegeben und sei o. B. d. A. $b \geq 0$.

Für die $m$ Nebenbedingungen führen wir **Hilfsvariablen** $h_1 \geq 0, \ldots, h_m \geq 0$ ein.
Die NB $a_i \cdot x = b_i$ ersetzen wir für jedes $i$ durch die NB $a_i \cdot x + \mathbf{h_i} = b_i$.
Wir ignorieren die Zielfunktion und definieren als **neue Zielfunktion** $h_1 + \ldots + h_m$.

**Zulässige Lösung für dieses LP:**

$h_i = b_i$ für jedes $i \in \{1, \ldots, m\}$ und $x_i = 0$ für alle $i \in \{1, \ldots, d\}$.

### Finden einer initialen Lösung

Sei ein LP mit den Nebenbedingungen $Ax = b$ gegeben und sei o. B. d. A. $b \geq 0$.

Für die $m$ Nebenbedingungen führen wir **Hilfsvariablen** $h_1 \geq 0, \ldots, h_m \geq 0$ ein.
Die NB $a_i \cdot x = b_i$ ersetzen wir für jedes $i$ durch die NB $a_i \cdot x + \mathbf{h_i} = b_i$.
Wir ignorieren die Zielfunktion und definieren als **neue Zielfunktion** $h_1 + \ldots + h_m$.

**Zulässige Lösung für dieses LP:**

$h_i = b_i$ für jedes $i \in \{1, \ldots, m\}$ und $x_i = 0$ für alle $i \in \{1, \ldots, d\}$.

Initialisiere Simplex-Algorithmus mit dieser Lösung und berechne eine opt. Lösung $(x^*, h^*)$.

Gilt $h^* \neq 0$, dann ist das ursprüngliche LP **nicht zulässig**.

### Finden einer initialen Lösung

Sei ein LP mit den Nebenbedingungen $Ax = b$ gegeben und sei o. B. d. A. $b \geq 0$.

Für die $m$ Nebenbedingungen führen wir **Hilfsvariablen** $h_1 \geq 0, \ldots, h_m \geq 0$ ein.
Die NB $a_i \cdot x = b_i$ ersetzen wir für jedes $i$ durch die NB $a_i \cdot x + \mathbf{h_i} = b_i$.
Wir ignorieren die Zielfunktion und definieren als **neue Zielfunktion** $h_1 + \ldots + h_m$.

**Zulässige Lösung für dieses LP:**

$h_i = b_i$ für jedes $i \in \{1, \ldots, m\}$ und $x_i = 0$ für alle $i \in \{1, \ldots, d\}$.

Initialisiere Simplex-Algorithmus mit dieser Lösung und berechne eine opt. Lösung $(x^*, h^*)$.

Gilt $h^* \neq 0$, dann ist das ursprüngliche LP **nicht zulässig**.

Gilt $h^* = 0$, dann ist $x^*$ eine **zulässige Lösung für das ursprüngliche LP**.