

## Lineares Programmieren

### Eingabe:

Eine Menge von linearen Ungleichungen  $S_i: \sum_{j=1}^m A_{ij}x_j \leq b_i$ .

Eine lineare Optimierungsfunktion  $f(x) = \sum_{j=1}^m c_j x_j$ .

### Problem:

Gesucht ist eine Belegung  $x \geq 0$ , für die alle Ungleichungen erfüllt sind und die  $f(x)$  maximiert.

**Beispiel**

Gegeben ein  $s$ - $t$ -Netzwerk  $G = (V, E)$ .

Maximiere  $\sum_{v \in V} f(s, v)$  unter den Bedingungen

- $f(u, v) = -f(v, u)$  für alle  $(u, v) \in E$ ,
- $\sum_{u \in V} f(v, u) = 0$  für alle  $v \in V - \{s, t\}$ ,
- $f(u, v) \leq c(u, v)$ .

⇒ Lineares Programm für das Maximale-Fluß-Problem.

## Modellierung als Lineares Programm

Maximiere  $c^T x$  unter  $Ax \leq b$  und  $x \geq 0$ .

Dabei ist  $x \in \mathbf{R}^n$ ,  $c \in \mathbf{R}^n$ ,  $b \in \mathbf{R}^m$  und  $A \in \mathbf{R}^{m \times n}$ .

Viele abweichende Probleme lassen sich in diese Form bringen:

- Minimiere  $c^T x$   $\mapsto$  Maximiere  $-c^T x$
- $a^T x \geq b$   $\mapsto$   $-a^T x \leq -b$
- Unbeschränktes  $x_i$   $\mapsto$   $x'_i - x''_i$  mit  $x'_i, x''_i \geq 0$

## Beispiel: Wassermischen

Wir brauchen 100 Liter Wasser mit  $60^{\circ}\text{C}$  und nehmen

- $x_1$  Liter mit  $20^{\circ}\text{C}$  zu 1 EUR/l
- $x_2$  Liter mit  $50^{\circ}\text{C}$  zu 2 EUR/l
- $x_3$  Liter mit  $100^{\circ}\text{C}$  zu 3 EUR/l

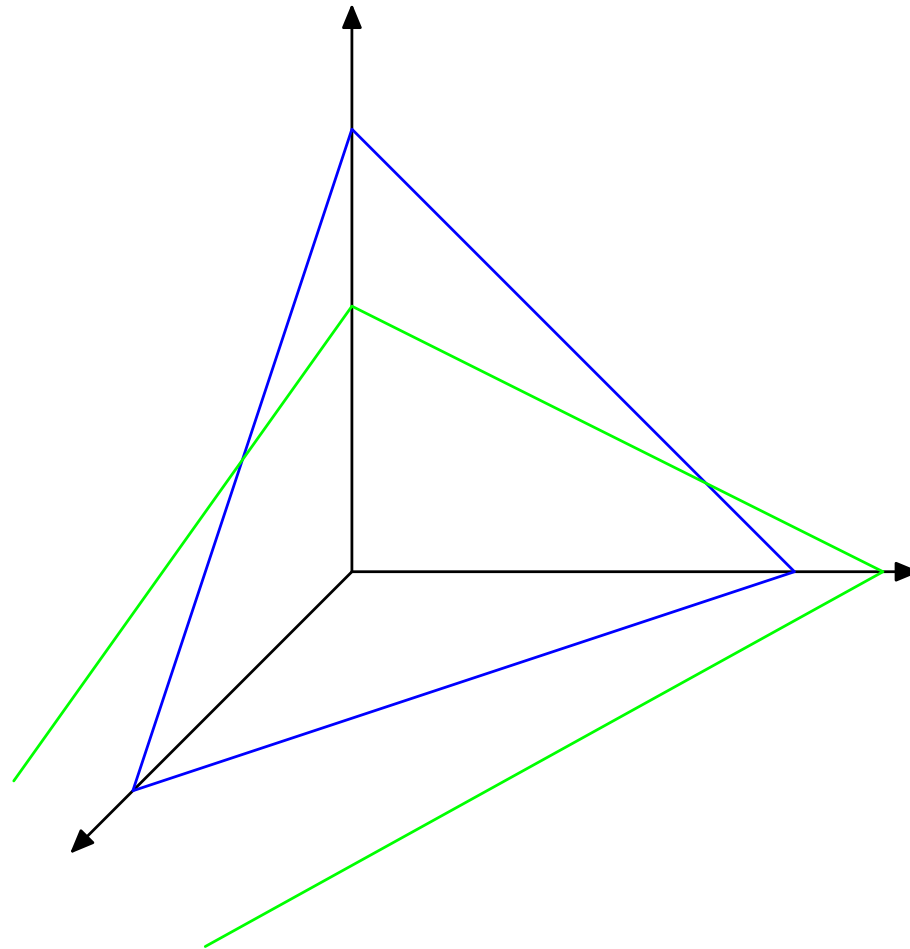
Minimiere  $x_1 + 2x_2 + 3x_3$  (die Kosten) unter

- $x_1 + x_2 + x_3 = 100$
- $20x_1 + 50x_2 + 100x_3 = 6000$

**Beispiel: Wassermischen**

$$x_1 + x_2 + x_3 = 100$$

$$20x_1 + 50x_2 + 100x_3 = 6000$$



## Kanonische und Normalform

*Kanonische Form:* Minimiere  $c^T x$  unter  $Ax \geq b$

*Normalform:* Minimiere  $c^T x$  unter  $Ax = b, x \geq 0$

Umwandlung Kanonisch  $\rightarrow$  Normalform:

Minimiere  $c_T x^+ - c^T x^-$  unter  $Ax^+ - Ax^- - Is = b, x^+, x^-, s \geq 0$

Umwandlung Normalform  $\rightarrow$  Kanonisch:

Minimiere  $c^T x$  unter  $\begin{pmatrix} A \\ -A \\ I \end{pmatrix} x \geq \begin{pmatrix} b \\ -b \\ 0 \end{pmatrix}$

**Beispiel**

$$x_1 + x_2 + x_3 = 100$$

$$20x_1 + 50x_2 + 100x_3 = 6000$$

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

---

$$x_1 + x_2 + x_3 \geq 100$$

$$-x_1 - x_2 - x_3 \geq -100$$

$$20x_1 + 50x_2 + 100x_3 \geq 6000$$

$$-20x_1 - 50x_2 - 100x_3 \geq -6000$$

$$x_1 \geq 0$$

$$x_2 \geq 0$$

$$x_3 \geq 0$$

**Beispiel**

$$x_1 + x_2 \geq 5$$

$$x_1 - 5x_2 \geq 6$$

$$3x_1 - x_2 \geq -2$$

---

$$x_1 + x_2 - x_3 = 5$$

$$x_1 - 5x_2 - x_4 = 6$$

$$3x_1 - x_2 - x_5 = -2$$

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

$x_3, x_4, x_5$  sind *Schlupfvariablen*.

Ist dies bereits Normalform?

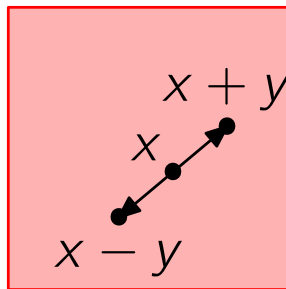


## Geometrie des Linearen Programmierens

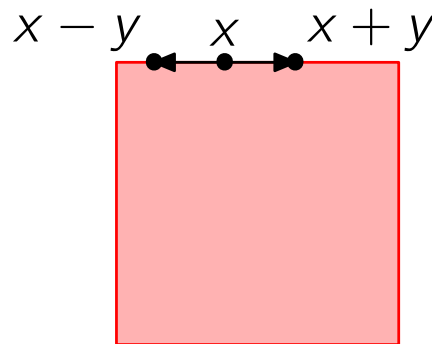
Sei  $P = \{x \mid Ax = b, x \geq 0\} \subseteq \mathbf{R}^n$ .

### Definition

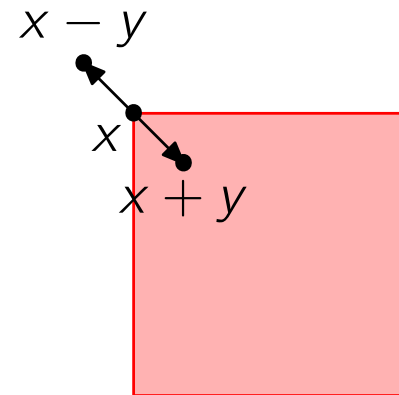
$x$  ist eine *Ecke* von  $P$ , wenn es **kein**  $y \neq 0$  mit  $x + y \in P$  und  $x - y \in P$  gibt.



keine Ecke



keine Ecke



Ecke!

## Geometrie des Linearen Programmierens

### Theorem A

Sei  $P = \{x \mid Ax = b, x \geq 0\}$ .

$\min\{c^T x \mid x \in P\}$  sei endlich (d.h., es existiere).

Dann gibt es für jedes  $x \in P$  eine Ecke  $x' \in P$  mit

$$c^T x' \leq c^T x.$$

### Korollar

Das Minimum wird (auch) in einer Ecke angenommen.

**Beweis**

Sei  $P = \{x \mid Ax = b, x \geq 0\}$ .

Wenn  $x \in P$  bereits eine Ecke ist, dann  $x' := x$ .

Anderenfalls gibt es ein  $y \neq 0$  mit  $x + y, x - y \in P$ .

Es gilt  $Ay = 0$ , denn  $A(x + y) = A(x - y) = b$ .

O.B.d.A. sei  $c^T y \leq 0$  (sonst nehme  $-y$ ).

Falls  $c^T y = 0$ , existiere ein  $j$  mit  $y_j < 0$  (sonst nehme  $-y$ ).

**Beweis** (Fortsetzung)

Wir zeigen nun, daß  $y \geq 0$  nicht gelten kann:

Wenn doch, dann  $c^T y < 0$ .

Dann ist  $x + \lambda y \in P$  für alle  $\lambda \geq 0$ , denn  $A(x + \lambda y) = b$  wegen  $Ay = 0$ .

Aber:  $c^T(x + \lambda y) = c^T x + \lambda c^T y \rightarrow -\infty$  für  $\lambda \rightarrow \infty$ .

Also ist  $\min\{c^T x \mid x \in P\}$  nicht endlich. Widerspruch.

**Beweis** (Fortsetzung)

Es gibt ein  $j$  mit  $y_j < 0$ .

- Wähle  $\lambda = \min \left\{ -\frac{x_j}{y_j} \mid y_j < 0 \right\} = -\frac{x_k}{y_k}$ .
- Dies ist das größte  $\lambda$  mit  $x + \lambda y \geq 0$ .
- $x + \lambda y \in P$ , denn  $Ay = 0 \Rightarrow A(x + \lambda y) = b$ .
- $(x + \lambda y)_k = 0$ , aber  $x_k > 0$ .

Ersetze jetzt  $x$  durch  $x + \lambda y$ .

Da es nur  $n$  Komponenten gibt, kann nach höchstens  $n$  Iterationen kein weiteres  $y$  mehr existieren und  $x$  muß eine Ecke sein.  $\square$

## Ecken und Basen

### Theorem B

Sei  $P = \{x \mid Ax = b, x \geq 0\}$  und  $x \in P$ . Mit  $A_x$  bezeichnen wir die Untermatrix von  $A$ , die aus den Spalten  $j$  besteht, für die  $x_j > 0$  gilt.

Dann ist  $x$  genau dann eine Ecke von  $P$ , wenn die Spalten von  $A_x$  linear unabhängig sind.

### Beispiel

$$A = \begin{pmatrix} 2 & 1 & 3 & 0 \\ 7 & 3 & 2 & 1 \\ 0 & 0 & 0 & 5 \end{pmatrix} \quad x = \begin{pmatrix} 2 \\ 0 \\ 1 \\ 0 \end{pmatrix} \quad A_x = \begin{pmatrix} 2 & 3 \\ 7 & 2 \\ 0 & 0 \end{pmatrix}$$

**Korollar** Es gibt nur endlich viele Ecken.

## Beweis

„ $\Leftarrow$ “

Nehmen wir an,  $x$  sei keine Ecke. Es gibt also ein  $y \neq 0$  mit  $x + y, x - y \in P$ . Sei  $A_y$  die Untermatrix, die zu den nicht-Null-Komponenten von  $y$  gehört.

- Aus  $A(x + y) = A(x - y) = b$  folgt wieder  $Ay = 0$ .  
Wegen  $y \neq 0$  hat  $A_y$  linear abhängige Spalten.
- $x_j = 0 \Rightarrow y_j = 0$ , denn  $x + y \geq 0$  und  $x - y \geq 0$ .  
Daher ist  $A_y$  eine Untermatrix von  $A_x$ .

Dann hat auch  $A_x$  linear abhängige Spalten.

**Beweis** (Fortsetzung)„ $\implies$ “

Nehmen wir an,  $A_x$  hat linear abhängige Spalten. Dann gibt es ein  $y \neq 0$  mit  $A_x y = 0$ .

Durch Erweiterung mit 0-Komponenten erhalten wir ein  $y \neq 0$  mit  $Ay = 0$  und

$$x_j = 0 \implies y_j = 0.$$

Für ein genügend kleines  $\epsilon > 0$  muß dann  $x + \epsilon y \in p$  und  $x - \epsilon y \in P$  gelten.

Also ist  $x$  keine Ecke!  $\square$



## Basen

Sei  $P = \{x \mid Ax = b, x \geq 0\}$ ,  $A \in \mathbf{R}^{m \times n}$ .

Für  $B \subseteq \{1, \dots, n\}$  sei  $A_B$  die Untermatrix von  $A$ , die aus allen Spalten in  $B$  besteht.

1. Falls  $B = \{j \mid x_j > 0\}$  für ein  $x \in P$ , dann  $A_B = A_x$ .
2. Falls  $x$  eine Ecke von  $P$  ist, dann ist  $A_x$  eine Matrix mit vollem Spaltenrang  $\leq m$ .
3. Wir nehmen ab jetzt an, daß  $\text{Rang}(A) = m$  gilt, denn sonst gibt es eine redundante Gleichung, die eliminiert werden kann.
4.  $A_x$  läßt sich immer zu einer quadratischen Matrix  $A_B$  mit  $m$  linear unabhängigen Spalten erweitern.
5. Eine quadratische Matrix  $A_B$  mit vollem Rang heißt *Basis*.

**Basen****Beispiel**

$$A = \begin{pmatrix} 2 & 1 & 3 & 0 \\ 7 & 3 & 2 & 1 \\ 0 & 0 & 0 & 5 \end{pmatrix} \quad x = \begin{pmatrix} 2 \\ 0 \\ 1 \\ 0 \end{pmatrix} \quad A_x = \begin{pmatrix} 2 & 3 \\ 7 & 2 \\ 0 & 0 \end{pmatrix}$$

$$A_B = \begin{pmatrix} 2 & 3 & 0 \\ 7 & 2 & 1 \\ 0 & 0 & 5 \end{pmatrix} \quad \text{mit } B = \{1, 3, 4\} \text{ ist eine Basis}$$

Sei  $A_B$  eine Basis und  $N = \{1, \dots, n\} - B$ .

Die Variablen in  $B$  heißen *Basisvariablen* und die Variablen in  $N$  heißen *Nicht-Basisvariablen*.

$Ax$  kann als  $A_B x_B + A_N x_N$  geschrieben werden und daher

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

für  $x \in P$  (denn  $A_B x_B + A_N x_N = b$ ).

Wenn  $x$  eine Ecke zu  $A_B$  ist, dann ist  $A_B x_B = b$  wegen  $x_N = 0$ .

Die Zielfunktion läßt sich schreiben als:

$$\begin{aligned} c^T x &= c_B^T x_B + c_N^T x_N \\ &= c_B^T (A_B^{-1}b - A_B^{-1}A_N x_N) + c_N^T x_N \\ &= c_B^T A_B^{-1}b + \underbrace{(c_N^T - c_B^T A_B^{-1}A_N)}_{\bar{c}^T} x_N \end{aligned}$$

## Der Simplex-Algorithmus

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

Falls  $\bar{c} \geq 0$ , dann ist  $x \in P$  eine optimale Lösung.

Sei  $x_B = A_B^{-1} b$  und  $x_N = 0$  die zu  $A_B$  gehörige Ecke. Insbesondere gelte  $x \in P$ , also  $x_B \geq 0$ .

Falls  $x_j > 0$  für alle  $j \in B$ , dann ist  $A_B$  *nicht entartet*, andernfalls *entartet*. Wir nehmen zunächst an, alle Basen seien nicht entartet.

Wähle  $j \in N$  mit  $\bar{c}_j < 0$

$$x_j := x_j + \lambda, \quad x_B := A_B^{-1} b - A_B^{-1} A_N x_N$$

für ein maximales  $\lambda$ , so daß  $x_B \geq 0$

Wähle  $k \in B$  mit  $x_k = 0$

$$B := B \cup \{j\} - \{k\}, \quad N := \{1, \dots, n\} - B$$

## Der Simplex-Algorithmus

Wähle  $j \in N$  mit  $\bar{c}_j < 0$

$$x_j := x_j + \lambda, x_B := A_B^{-1}b - A_B^{-1}A_N x_N$$

für ein maximales  $\lambda$ , so daß  $x_B \geq 0$

Wähle  $k \in B$  mit  $x_k = 0$

$$B := B \cup \{j\} - \{k\}, N := \{1, \dots, n\} - B$$

Wir erhalten eine neue Basis  $A_B$  mit einem neuen Vektor  $x$  und einer **kleineren Zielfunktion**  $c^T x$ :

$$c^T x = c_B^T A_B^{-1}b + \bar{c}^T x_N$$

Da es nur endlich viele Basen gibt, kann man durch Wiederholen in endlicher Zeit eine optimale Lösung finden.

**Beispiel**

Minimiere  $c^T x$  unter  $Ax = b$  und  $x \geq 0$  mit

$$A = \begin{pmatrix} 2 & 1 & 3 & 0 \\ 7 & 3 & 2 & 1 \\ 0 & 0 & 0 & 5 \end{pmatrix} \quad b = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} \quad c = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

Wir wählen folgende Basis  $A_B$ :

$$A_B = \begin{pmatrix} 2 & 3 & 0 \\ 7 & 2 & 1 \\ 0 & 0 & 5 \end{pmatrix} \quad A_B^{-1} = \frac{1}{85} \begin{pmatrix} -10 & 15 & -3 \\ 35 & -10 & 2 \\ 0 & 0 & 17 \end{pmatrix} \quad x_B = \frac{1}{85} \begin{pmatrix} 11 \\ 21 \\ 51 \end{pmatrix}$$

**Beispiel**

$$A_B = \begin{pmatrix} 2 & 3 & 0 \\ 7 & 2 & 1 \\ 0 & 0 & 5 \end{pmatrix} \quad A_B^{-1} = \frac{1}{85} \begin{pmatrix} -10 & 15 & -3 \\ 35 & -10 & 2 \\ 0 & 0 & 17 \end{pmatrix} \quad x_B = \frac{1}{85} \begin{pmatrix} 11 \\ 21 \\ 51 \end{pmatrix}$$

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

$$1 - \begin{pmatrix} 1 & 1 & 1 \end{pmatrix} \frac{1}{85} \begin{pmatrix} -10 & 15 & -3 \\ 35 & -10 & 2 \\ 0 & 0 & 17 \end{pmatrix} \begin{pmatrix} 1 \\ 3 \\ 0 \end{pmatrix} = \frac{9}{17}$$

Da  $\bar{c} \geq 0$  ist  $x = \frac{1}{85} \begin{pmatrix} 11 & 21 & 51 & 0 \end{pmatrix}^T$  optimal.

## Pivotieren

Falls  $A_B$  entartet ist, kann es ein Problem geben: Obwohl  $\bar{c}_j < 0$ , können wir  $x_j$  nicht erhöhen, ohne eine Komponente von  $x_B$  negativ zu machen.

Dies kommt in der Praxis oft vor.

Wir können dann *pivotieren*: Wir wählen ein  $k \in B$  mit  $x_k = 0$  und ein  $j \in N$ . Dann setzen wir

$$B := B - \{k\} \cup \{j\} \text{ und } N := N - \{j\} \cup \{k\}.$$

Dabei ändert sich  $x$  und  $c^T x$  nicht, aber wir haben eine andere Basis.

Es kann zu einem Zyklus kommen und der Algorithmus terminiert nicht.

Das kommt in der Praxis nicht oft vor.



## Wahl des Pivot-Elements

Zusammenfassung:

- Wir können die  $s$ -te Spalte als Pivotzeile wählen, falls  $\bar{c}_s < 0$ .
- Eine übliche Wahl ist das minimale  $\bar{c}_s$  (steepest descent).
- Falls  $\bar{a}_{is} \leq 0$  für alle  $i \in \{1, \dots, m\}$ , dann ist die minimale Lösung unbeschränkt.
- Andernfalls wählt man  $t$  als Pivotspalte, so daß  $\bar{b}_t/\bar{a}_{ts}$  minimal für  $\bar{a}_{ts} > 0$  ist.
- Falls es mehrere gleich gute  $\bar{b}_t/\bar{a}_{ts}$  gibt, dann gelangt man zu einer entarteten Basis. Durch eine *Pivotstrategie* muß eines der  $t$  gewählt werden.

## Pivotstrategien

Eine übliche Strategie ist es, nachdem  $s$  durch steepest-descent festgelegt wurde, unter den möglichen  $t$  das **erste** zu nehmen.

Dabei kann es aber zu einem Zyklus kommen, der aus einer entarteten Ecke nicht mehr herauskommt. Dies passiert aber in der Praxis so gut wie nie.

Durch zufällige Wahl eines Kandidaten läßt sich dieses pathologische Verhalten einfach verhindern.

Es gibt deterministische Strategien, die einen Zyklus ausschließen. Sie sind aber umständlich zu implementieren und scheinen langsamer zur optimalen Lösung zu gelangen.

## Der Zweiphasenalgorithmus

Es sei ein Problem in der Form  $Ax \leq b, x \geq 0$  mit  $b \geq 0$  gegeben.

Dann ist das zugehörige Problem in Standardform

$$Ax + Iy = b, x \geq 0, y \geq 0.$$

Dafür ist  $y = b, x = 0$  eine zulässige Basislösung, falls wir  $y$  als Basisvariablenvektor wählen. Wir können sofort mit dem Simplexalgorithmus starten.

Was machen wir aber, wenn wir ein allgemeines Problem lösen wollen, für das eine zulässige Basislösung nicht bekannt ist?

## Der Zweiphasenalgorithmus

Es sei  $c^T x$  zu minimieren unter  $Ax = b, x \geq 0$  (Problem A).

Konstruiere ein neues Problem (Problem B):

Minimiere  $s_1 + \dots + s_m$  unter  $Ax + Is = b, x \geq 0, y \geq 0$ .

### Theorem

Es gibt eine zulässige Basislösung zu Problem A genau dann, wenn  $s = 0$  die optimale Lösung von Problem B ist. Ist  $(x, s)$  die Lösung von B, dann ist  $x$  eine zulässige Basislösung von A.

## Der Zweiphasenalgorithmus

Es sei  $c^T x$  zu minimieren unter  $Ax = b, x \geq 0$ .

1. Negiere alle Zeilen  $i$  mit  $b_i < 0$ .
2. Minimiere  $s_1 + \dots + s_m$  unter  $Ax + Is = b, x \geq 0, y \geq 0$  mit dem Simplexalgorithmus. Nehme  $x = 0, s = b$  als zulässige Anfangsbasislösung.
3. Falls  $s_1 + \dots + s_m > 0$ , dann gibt es keine Lösung zum Ursprungsproblem.
4. Sonst ist jetzt ein  $x \geq 0$  bekannt, mit  $Ax = b$ .
5. Aus diesem  $x$  ergibt sich eine zulässige Basislösung.
6. In der zweiten Phase kann jetzt das ursprüngliche Problem mit dem Simplexalgorithmus gelöst werden.

## Dualität

*Dualität* beschäftigt sich mit Paaren von linearen Programmen und der Beziehung zwischen ihnen.

Das *primale Problem* sei

$$z = \max\{ c^T x \mid Ax \leq b, x \geq 0 \}. \quad (P)$$

Das zugehörige *duale Problem* ist dann definiert als

$$w = \min\{ b^T u \mid A^T u \geq c, u \geq 0 \} \quad (D)$$

oder, anders geschrieben,

$$w = \min\{ u^T b \mid u^T A \geq c^T, u \geq 0 \}.$$

**Beispiel**

$$\begin{array}{rcllcl} \text{Max} & 7x_1 & + & 2x_2 & & \\ & - & x_1 & + & 2x_2 & \leq 4 \\ & & 5x_1 & + & x_2 & \leq 20 \\ & - & 2x_1 & - & 2x_2 & \leq -7 \\ & & & & x_1, x_2 & \geq 0 \end{array}$$

---

$$\begin{array}{rcllcl} \text{Min} & 4u_1 & + & 20u_2 & - & 7u_3 \\ & - & u_1 & + & 5u_2 & - & 2u_3 & \geq 7 \\ & & 2u_1 & + & u_2 & - & 2u_3 & \geq 2 \\ & & & & u_1, u_2, u_3 & \geq 0 \end{array}$$

## Dualität

### Theorem

Das duale Problem des dualen Problems ist wieder das primale.

**Beweis** Wir beginnen mit dem dualen Problem:

$$\begin{aligned} w &= \min\{ u^T b \mid u^T A \geq c^T, u \geq 0 \} \\ &= \max\{ -b^T x \mid -A^T x \leq -c, x \geq 0 \} \end{aligned}$$

Das duale Problem hiervon ist:

$$\begin{aligned} &= \min\{ -c^T x \mid -Au \geq -b, u \geq 0 \} \\ &= \max\{ c^T x \mid Ax \leq b, x \geq 0 \} \end{aligned}$$

Das ist jetzt das primale Problem.  $\square$



## Schwache Dualität

Sei wieder

$$z = \max\{ c^T x \mid Ax \leq b, x \geq 0 \} \quad (P)$$

und

$$w = \min\{ b^T u \mid A^T u \geq c, u \geq 0 \}. \quad (D)$$

$x$  ist *primal zulässig*, wenn  $x \in \{ x \mid Ax \leq b, x \geq 0 \}$ .

$u$  ist *dual zulässig*, wenn  $u \in \{ u \mid u^T A \geq c^T, u \geq 0 \}$ .

**Theorem** (Schwache Dualität)

Sei  $\hat{x}$  primal zulässig und  $\hat{u}$  dual zulässig. Dann gilt

$$c^T \hat{x} \leq z \leq w \leq b^T \hat{u}.$$

## Schwache Dualität

### Beweis

$$c^T \hat{x} \leq \hat{u}^T A \hat{x} \leq b^T \hat{u}, \text{ denn}$$

1.  $\hat{u}^T A \geq c^T$  und  $\hat{x} \geq 0$
2.  $A \hat{x} \leq b$  und  $\hat{u} \geq 0$

Da  $z = c^T x$  für ein primal zulässiges  $x$  und  $w = b^T u$  für ein dual zulässiges  $u$ , folgt  $z \leq w$ .

### Korollar

Falls  $(P)$  eine unbeschränkte optimale Lösung hat, dann hat  $(D)$  keine Lösung.

## Dualität

Folgende Paare sind dual:

$$z = \min\{ c^T x \mid Ax = b, x \geq 0 \} \quad (P)$$

$$w = \max\{ b^T y \mid A^T y \leq c \} \quad (D)$$

**Beweis:**

$$\begin{aligned} z &= \min\{ c^T x \mid Ax = b, x \geq 0 \} \\ &= \min\left\{ c^T x \mid \begin{pmatrix} A \\ -A \end{pmatrix} x \geq \begin{pmatrix} b \\ -b \end{pmatrix}, x \geq 0 \right\} \end{aligned}$$

$$\begin{aligned} z &= \min \{ c^T x \mid Ax = b, x \geq 0 \} \\ &= \min \left\{ c^T x \mid \begin{pmatrix} A \\ -A \end{pmatrix} x \geq \begin{pmatrix} b \\ -b \end{pmatrix}, x \geq 0 \right\} \end{aligned}$$

Das duale Problem ist nach Definition:

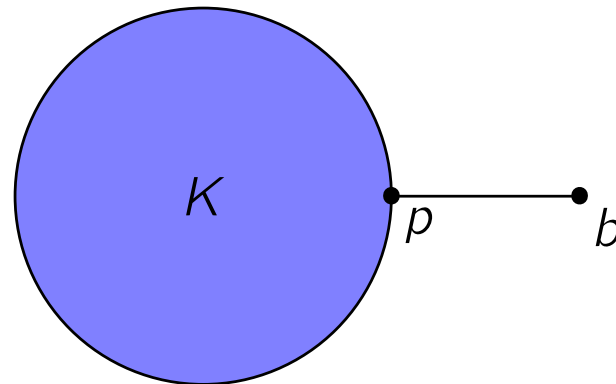
$$\begin{aligned} w &= \max \left\{ (b^T, -b^T) \begin{pmatrix} y \\ \bar{y} \end{pmatrix} \mid (A^T, -A^T) \begin{pmatrix} y \\ \bar{y} \end{pmatrix} \leq c, y, \bar{y} \geq 0 \right\} \\ &= \max \{ b^T (y - \bar{y}) \mid A^T (y - \bar{y}) \leq c, y, \bar{y} \geq 0 \} \\ &= \max \{ b^T y \mid A^T y \leq c \} \end{aligned}$$

□

**Lemma** (Das Projektionstheorem)

Sei  $K$  eine abgeschlossene, konvexe Teilmenge des  $\mathbf{R}^n$  und  $b \in \mathbf{R}^n$ .

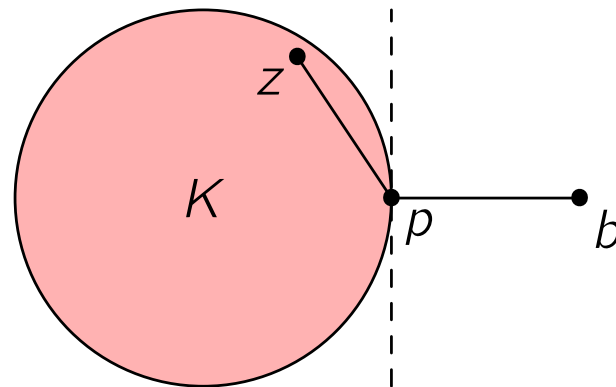
Sei  $p$  die *Projektion* von  $b$  auf  $K$  (d.h. der Punkt in  $K$  mit minimalem Euklidischen Abstand  $|p - b|$  von  $b$ ).



Dann gilt

$$(z - p)^T (b - p) \leq 0$$

für alle  $z \in K$ .

**Beweis**

Da  $K$  konvex ist, liegen  $z$  und  $b$  auf verschiedenen Seiten der Hyperebene, die normal zu  $b - p$  durch  $p$  geht (wobei  $z$  auch auf dieser Hyperebene liegen kann).

Sei  $\alpha$  der Winkel zwischen  $z - p$  und  $b - p$ . Dann ist  $\alpha \geq \pi/2$ .

$$\Rightarrow (z - p)^T (b - p) = |z - p| \cdot |b - p| \cos \alpha \leq 0.$$

(Falls  $b \in K$ , dann gilt die Aussage trivialerweise.)

## Dualität

### Theorem (Farkas Lemma A)

Sei  $A \in \mathbf{R}^{m \times n}$ ,  $b \in \mathbf{R}^m$ . Dann ist genau eine dieser Aussagen wahr:

1. Es gibt ein  $x \in \mathbf{R}^n$  mit  $Ax = b$ ,  $x \geq 0$ .
2. Es gibt ein  $y \in \mathbf{R}^m$  mit  $A^T y \geq 0$ ,  $b^T y < 0$ .

### Beweis

Angenommen beide Aussagen gälten.

$$Ax = b \Rightarrow y^T Ax = y^T b \Rightarrow x^T A^T y = y^T b < 0 \quad (1)$$

$$x \geq 0, A^T y \geq 0 \Rightarrow x^T A^T y \geq 0 \quad (2)$$

Offensichtlich widersprechen sich (1) und (2). Also kann höchstens eine der Aussagen wahr sein.

Nehmen wir jetzt an, die **erste Aussage** gilt **nicht**.

Sei  $K = \{ Ax \mid x \geq 0 \}$ .  $K$  ist abgeschlossen und konvex und  $b \notin K$ .

Sei  $p$  die Projektion von  $b$  auf  $K$ .

Es gibt ein  $w$  mit  $Aw = p$ .

Gemäß Projektionstheorem gilt dann

$$(z - p)^T (b - p) \leq 0 \text{ für alle } z \in K.$$

Insbesondere gilt

$$(Ax - p)^T (b - p) \leq 0 \text{ für alle } x \geq 0.$$



Wir wählen  $y = p - b$ . Dann gilt

$$(Ax - p)^T (b - p) \leq 0 \text{ für alle } x \geq 0$$

$$\Rightarrow (Ax - p)^T y \geq 0 \text{ für alle } x \geq 0$$

$$\Rightarrow (Ax - Aw)^T y \geq 0 \text{ für alle } x \geq 0$$

$$\Rightarrow (x - w)^T A^T y \geq 0 \text{ für alle } x \geq 0$$

$$\Rightarrow x^T A^T y \geq 0 \text{ für alle } x \geq 0 \text{ (denn } w \geq 0)$$

$$\Rightarrow A^T y \geq 0$$

Damit haben wir  $A^T y \geq 0$  bewiesen.

Wir haben  $A^T y \geq 0$  und brauchen noch  $b^T y < 0$ .

$$y^T b = (p - y)^T y = p^T y - y^T y$$

Es gilt aber

$$(Ax - p)^T y \geq 0$$

für alle  $x \geq 0$  (vorige Folie).

Insbesondere ist  $p^T y \leq 0$  (aus  $x = 0$ ).

Es ist  $y^T y > 0$ , denn  $y = p - b \neq 0$ .

$$\Rightarrow y^T b = p^T y - y^T y < 0$$

□

## Dualität

**Theorem** (Farkas Lemma B)

Sei  $A \in \mathbf{R}^{m \times n}$ ,  $b \in \mathbf{R}^m$ . Dann ist genau eine dieser Aussagen wahr:

1. Es gibt ein  $x \in \mathbf{R}^n$  mit  $Ax \leq b$ .
2. Es gibt ein  $y \geq 0$  mit  $A^T y = 0$ ,  $b^T y < 0$ .

**Beweis**

Übungsaufgabe!

## Starke Dualität

$$z = \min\{ c^T x \mid Ax = b, x \geq 0 \} \quad (P)$$

$$w = \max\{ b^T y \mid A^T y \leq c \} \quad (D)$$

### Theorem (Starke Dualität)

Falls das primale oder duale Problem zulässige Lösungen besitzt, dann gilt  $z = w$ .

### Beweis

Wir müssen lediglich  $z \leq w$  zeigen.

O.B.d.A. habe  $P$  eine zulässige Lösung (Dualität!).

Falls  $P$  unbeschränkt ist, dann  $z = w = -\infty$  wegen schwacher Dualität.

**Beweis** (Fortsetzung)

Sei nun  $P$  beschränkt und  $x^*$  eine optimale primale Lösung. Es gilt daher  $Ax^* = b$  und  $c^T x^* = z$ .

Wenn wir zeigen können, daß es ein  $y$  gibt, mit

$$A^T y \leq c \text{ und } b^T y \geq z,$$

dann sind wir fertig, denn  $y$  ist eine zulässige duale Lösung. Der Wert der Zielfunktion ist mindestens  $z$ . Also ist  $w \geq z$ .

Nehmen wir also an, so ein  $y$  existiere nicht.

Wir verwenden jetzt diese Form von Farkas Lemma:

1. Entweder  $\exists \hat{x}$  mit  $\hat{A}\hat{x} \leq \hat{b}$ ,
2. oder  $\exists \hat{y} \geq 0$  mit  $\hat{A}^T \hat{y} = 0$  und  $\hat{b}^T \hat{y} < 0$ .

Wir definieren:

$$\hat{A} := \begin{pmatrix} A^T \\ -b^T \end{pmatrix}, \quad \hat{b} := \begin{pmatrix} c \\ -z \end{pmatrix}, \quad \hat{x} := y, \quad \begin{pmatrix} x \\ \lambda \end{pmatrix} := \hat{y}$$

Nach unserer Annahme ist dann 1. im Lemma falsch und 2. muß wahr sein.

Die Annahme war: Es gibt kein  $y$  mit  $A^T y \leq c$  und  $b^T y \geq z$ .

Es gilt also 2.:  $\exists \hat{y} \geq 0$  mit  $\hat{A}^T \hat{y} = 0$  und  $\hat{b}^T \hat{y} < 0$ , wobei

$$\hat{A} := \begin{pmatrix} A^T \\ -b^T \end{pmatrix}, \quad \hat{b} := \begin{pmatrix} c \\ -z \end{pmatrix}, \quad \hat{x} := y, \quad \begin{pmatrix} x \\ \lambda \end{pmatrix} := \hat{y}.$$

Das bedeutet aber gerade, daß es  $x \geq 0$  und  $\lambda \geq 0$  gibt mit

$$Ax = \lambda b \text{ und } c^T x < \lambda z.$$

Falls  $\lambda > 0$ , dann gilt  $Ax' = b$  und  $c^T x' < z$  mit  $x' = \frac{1}{\lambda}x$ . Das ist ein Widerspruch zur Minimalität von  $z$ .

Falls aber  $\lambda = 0$ , dann ist  $A(x^* + x) = b$  und  $c^T(x^* + x) = z + c^T x < z$ . Das ist auch ein Widerspruch zur Minimalität von  $z$ .  $\square$

## Die Größe eines Linearen Programms

Sei  $k \in \mathbf{Z}$ ,  $x \in \mathbf{Z}^m$ ,  $M \in \mathbf{Z}^{m \times n}$ .

Wir definieren die *Größe einer Kodierung* als

$$\text{size}(k) := 1 + \lceil \log_2(|k| + 1) \rceil$$

$$\text{size}(x) := \sum_{i=1}^m \text{size}(x_i)$$

$$\text{size}(M) := \sum_{i=1}^m \sum_{j=1}^n \text{size}(M_{ij})$$



## Die Größe eines Linearen Programms

Sei  $A \in \mathbf{Z}^{m \times n}$ ,  $b \in \mathbf{Z}^m$ ,  $c \in \mathbf{Z}^n$ .

Minimiere  $c^T x$

unter  $Ax = b$  ( $LP$ )

$x \geq 0$

*Größe* dieses linearen Programms:

$$size(LP) := size(A) + size(b) + size(c)$$

(Durch Skalieren mit dem lcp läßt sich jedes LP in diese Form bringen.)

## Die Größe eines Linearen Programms

Statt mit  $size(LP)$  rechnet es sich oft günstiger mit einer Größe  $L$ , die so definiert ist:

$$L := size(det_{\max}) + size(b_{\max}) + size(c_{\max}) + m + n,$$

wobei

$$det_{\max} = \max_{A'} |det A'| \quad (A' \text{ Untermatrix von } A)$$

$$b_{\max} = \max_i |b_i|$$

$$c_{\max} = \max_i |c_i|$$

**Behauptung:**

$$L < size(LP)$$

$\Rightarrow$  Ist die Laufzeit polynomiell in  $L$ , dann auch in  $size(LP)$ .

## Größe der Ausgabe

Sei  $x$  eine Ecke von  $\{x \mid Ax = b, x \geq 0\}$ .

Dann ist

$$x^T = \left( \frac{p_1}{q} \quad \frac{p_2}{q} \quad \dots \quad \frac{p_n}{q} \right),$$

mit

$$p_i, q \in \mathbf{N}, 0 \leq p_i < 2^L, 1 \leq q < 2^L.$$

Die Ausgabe ist also **polynomiell repräsentierbar**.

## Interior Point Methods

### Theorem

Seien  $x_1$  und  $x_2$  Ecken von  $\{x \mid Ax = b, x \geq 0\}$ .

Falls  $c^T x_1 \neq c^T x_2$ , dann ist  $|c^T x_1 - c^T x_2| > 2^{-2L}$ .

### Korollar

Sei  $z = \min\{c^T x \mid Ax = b, x \geq 0\}$  und sei  $x$  eine zulässige Lösung mit  $c^T x \leq z + 2^{-2L}$ .

Dann ist jede Ecke  $x'$  mit  $c^T x' \leq c^T x$  eine optimale Lösung.

⇒ Es genügt nahe an eine Lösung heranzukommen und man muß nur mit einer beschränkten Genauigkeit rechnen.

## Interior Point Methods

Wir lösen gleichzeitig dieses primale und sein duales Problem:

$$\begin{aligned} &\text{Minimiere } z = c^T x \\ &\text{unter } Ax = b \\ &\quad x \geq 0 \end{aligned} \tag{P}$$

$$\begin{aligned} &\text{Maximiere } w = b^T y \\ &\text{unter } A^T y + s = c \\ &\quad s \geq 0 \end{aligned} \tag{D}$$

Der Algorithmus arbeitet mit einer primalen Lösung  $\bar{x} > 0$  und einer Belegung  $\bar{s} > 0$  von  $s$ , zu welcher es ein  $\bar{y}$  mit  $A^T \bar{y} + \bar{s} = c$  gibt.

## Interior Point Methods

Eine Grundidee des Algorithmus ist es im **Inneren** des Polyeders zu bleiben. Keine Komponente von  $\bar{x} > 0$  und  $\bar{s} > 0$  soll also klein werden.

Folgende Abbildung transformiert das Problem in einen Bildraum:

$$x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \mapsto x' = \begin{pmatrix} x_1/\bar{x}_1 \\ x_2/\bar{x}_n \\ \vdots \\ x_n/\bar{x}_n \end{pmatrix}$$

Diese Abbildung bildet  $\bar{x}$  auf  $e$  (den Einsvektor) ab.

## Skalierung

Wir können diese Abbildung auch so beschreiben:

$$x \mapsto x' = \bar{X}^{-1}x$$

mit

$$\bar{X} = \begin{pmatrix} \bar{x}_1 & 0 & 0 & \cdots & 0 \\ 0 & \bar{x}_2 & 0 & \cdots & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & \cdots & \bar{x}_{n-1} & 0 \\ 0 & 0 & \cdots & 0 & \bar{x}_{n-1} \end{pmatrix} .$$

## Skalierung

Im Bildraum sehen die Probleme so aus:

$$\begin{aligned} \text{Minimiere } z &= \bar{c}^T x' \\ \text{unter } \bar{A}x' &= b \\ x' &\geq 0 \end{aligned} \tag{P}$$

$$\begin{aligned} \text{Maximiere } w &= b^T y \\ \text{unter } \bar{A}^T y + s' &= \bar{c} \\ s' &\geq 0 \end{aligned} \tag{D}$$

Dabei ist  $\bar{A} = A\bar{X}$  und  $\bar{c} = \bar{X}c$ .



## Die Dualitätslücke

Es stellt sich heraus, daß

$$s' = \bar{X}s = \begin{pmatrix} s_1 \bar{x}_1 \\ s_2 \bar{x}_2 \\ \vdots \\ s_n \bar{x}_n \end{pmatrix}.$$

Die *Dualitätslücke*  $c^T \bar{x} - b^T \bar{y} = \bar{x}^T \bar{s}$  ändert sich durch Skalierung nicht: Es gilt ja  $x^T s = x'^T s'$ .

(Sobald die Dualitätslücke sehr klein ist, kann der Algorithmus terminieren.)

## Die Potentialfunktion

### Definition

$$G(x, s) = (n + \sqrt{n}) \ln(x^T s) - \sum_{j=1}^n \ln(x_j s_j)$$

1. Die Potentialfunktion ist unter Skalierung invariant
2. Man kann leicht einen zulässigen Startwert  $\bar{x} > 0$ ,  $\bar{s} > 0$  finden, mit  $G(\bar{x}, \bar{s}) = O(\sqrt{n}L)$
3. Ist  $G(\bar{x}, \bar{s}) \leq -2\sqrt{n}L$  dann sind wir nah am Optimum und können stoppen.
4. Wir können stets einen *Schritt*  $\bar{x}, \bar{s} \mapsto \tilde{x}, \tilde{s}$  machen, so daß  $G(\tilde{x}, \tilde{s}) - G(\bar{x}, \bar{s}) \leq -7/120$ .

⇒ Nach  $O(\sqrt{n}L)$  Iterationen sind wir fertig.

## Der primale Schritt

Wir machen entweder einen *primalen* oder einen *dualen Schritt*.

Der primale Schritt ändert nur  $\bar{x}$  und sieht so aus:

Zunächst gehen wir in den Bildraum (unser Punkt ist jetzt  $(e, s')$ ).

Jetzt möchten wir in eine Richtung gehen, in der die Potentialfunktion um möglichst viel kleiner wird.

Dazu betrachten wir den **Gradienten** von  $G(x, s)$  am Punkt  $(e, s')$ :

$$\begin{aligned} g &= \nabla_x G(x, s)|_{(e, s')} \\ &= \frac{n + \sqrt{n}}{x^T s} s - \begin{pmatrix} 1/x_1 \\ \vdots \\ 1/x_n \end{pmatrix} \Big|_{(e, s')} \\ &= \frac{n + \sqrt{n}}{e^T s'} s' - e \end{aligned}$$

Wir können aber nicht in Richtung  $-g$  gehen, da wir den zulässigen Bereich verlassen könnten.

Sei  $d$  die Projektion von  $g$  auf  $\{x \mid \bar{A}x = 0\}$ .

Wir können in Richtung  $-d$  gehen!

## Der primale Schritt

Der Vektor  $d$  läßt sich so ausdrücken:

$$d = (I - \bar{A}(\bar{A}\bar{A}^T)^{-1}\bar{A})g$$

Der primale Schritt geht zu

$$\tilde{x} = e - \frac{1}{4|d|}d, \quad \tilde{s} = s'.$$

(Danach muß wieder zurückskaliert werden.)

Um zu garantieren, daß die Potentialfunktion genügend abnimmt (um  $7/120$ ), muß  $|d| = \sqrt{d^T d} \geq 0.4$  gelten. Nur wenn dies der Fall ist, machen wir einen primalen Schritt.

## Der duale Schritt

Falls  $|d| < 0.4$  ist, machen wir einen *dualen Schritt*, der nur  $s'$  verändert. Der Gradient ist jetzt:

$$h = \nabla_s G(x, s)|_{(e, s')} = \frac{n + \sqrt{n}}{e^T s'} e - \begin{pmatrix} 1/s'_1 \\ \vdots \\ 1/s'_n \end{pmatrix}$$

Wir brauchen: Es muß ein  $y$  mit  $\bar{A}^T y + \tilde{s} = \bar{c}$  geben.

Es geht, wenn wir den Schritt so wählen:

$$\tilde{s} = \frac{e^T s'}{n + \sqrt{n}} (d + e), \quad \tilde{x} = x' = e$$

Man kann zeigen, daß  $\tilde{s} > 0$  und daß die Potentialfunktion um mindestens  $1/6$  abnimmt.

## Ganzzahliges Programmieren (ILP)

Sei  $A$  eine rationale Matrix und  $b, c$  rationale Vektoren. Das *ILP* ist es, folgende Optimierungsaufgaben zu lösen:

$$\begin{aligned} &\text{Maximiere } c^T x \\ &\text{unter } Ax \leq b \\ &\quad x \text{ ganzzahlig} \end{aligned}$$

oder

$$\begin{aligned} &\text{Maximiere } c^T x \\ &\text{unter } Ax = b \\ &\quad x \geq 0 \\ &\quad x \text{ ganzzahlig} \end{aligned}$$

## Eine einfache Abschätzung

Dualität liefert:

$$\begin{aligned} & \max\{ c^T x \mid Ax \leq b, x \text{ ganzzahlig} \} \\ & \leq \min\{ b^t y \mid A^T y = c, y \geq 0, y \text{ ganzzahlig} \} \end{aligned}$$

Im Gegensatz zu linearem Programmieren, ist die Ungleichung bei ILP normalerweise **echt**. Durch die *LP-Relaxation* erhalten wir eine Abschätzung, falls die Voraussetzung für starke Dualität vorliegt:

$$\begin{aligned} & \max\{ c^T x \mid Ax \leq b, x \text{ ganzzahlig} \} \\ & \leq \max\{ c^T x \mid Ax \leq b \} = \min\{ b^t y \mid A^T y = c, y \geq 0 \} \\ & \leq \min\{ b^t y \mid A^T y = c, y \geq 0, y \text{ ganzzahlig} \} \end{aligned}$$



**Beispiel**

Sei  $A = (2)$ ,  $b = (1)$ ,  $c = (1)$ .

$$\begin{aligned} & \max\{ c^T x \mid Ax \leq b, x \text{ ganzzahlig} \} && = 0 \\ & \leq \max\{ c^T x \mid Ax \leq b \} && = 1/2 \\ & = \min\{ b^t y \mid A^T y = c, y \geq 0 \} && = 1/2 \\ & \leq \min\{ b^t y \mid A^T y = c, y \geq 0, y \text{ ganzzahlig} \} && = \text{existiert nicht} \end{aligned}$$

**Beispiel**

Sei  $A = (2)$ ,  $b = (1)$ ,  $c = (1)$ .

$$\begin{aligned} & \max\{c^T x \mid Ax \leq b, x \geq 0, x \text{ ganzzahlig}\} && = 0 \\ & \leq \max\{c^T x \mid Ax \leq b, x \geq 0\} && = 1/2 \\ & = \min\{b^t y \mid A^T y \geq c, y \geq 0\} && = 1/2 \\ & \leq \min\{b^t y \mid A^T y \geq c, y \geq 0, y \text{ ganzzahlig}\} && = 1 \end{aligned}$$

## Komplexität von ILP

### Theorem

ILP ist  $NP$ -vollständig

### Beweis

Reduktion von 3SAT auf ILP.

Gegeben ist eine Instanz  $F$  von 3SAT bestehend aus  $n$  Variablen  $x_1, \dots, x_n$  und  $m$  Klauseln der Form  $\{l_1, l_2, l_3\}$ , wobei  $l_i \in \{x_1, \bar{x}_1, \dots, x_n, \bar{x}_n\}$ .

Wir konstruieren aus  $F$  ein ILP  $I$  mit:

$F$  hat eine erfüllende Belegung  $\iff$  der optimale Wert von  $I$  ist 0

Unser ILP  $I$  hat die Literale von  $F$  als Variablen und sieht so aus:

Maximiere 0

unter folgenden Bedingungen:

$$l_1 + l_2 + l_3 \geq 1 \text{ für alle Klauseln } \{l_1, l_2, l_3\} \text{ in } F$$

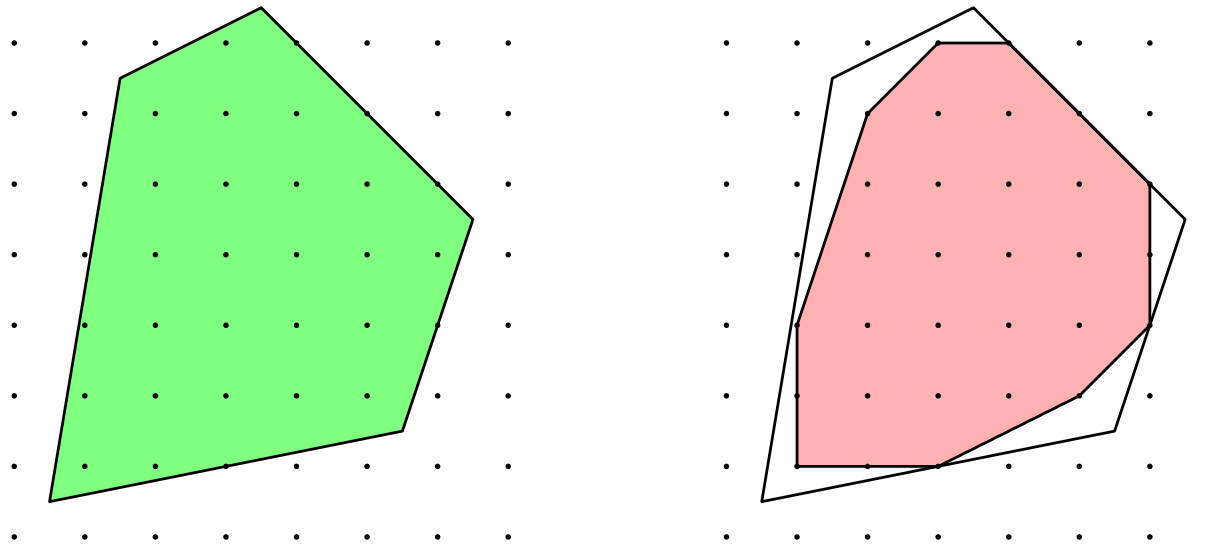
$$0 \leq x_i, \bar{x}_i \leq 1 \text{ für } i = 1, \dots, n$$

$$x_i + \bar{x}_i = 1 \text{ für } i = 1, \dots, n$$

$$x_1, \dots, x_n \text{ ganzzahlig}$$

Es gibt genau dann eine Lösung, wenn  $F$  erfüllbar ist. Die Konstruktion ist in polynomieller Zeit durchführbar.  $\square$

## Die ganzzahlige Hülle



Die *ganzzahlige Hülle*  $P_I$  eines Polyeders  $P$  ist die konvexe Hülle aller ganzzahliger Punkte in  $P$ .

Falls **alle** Ecken von  $P$  ganzzahlig sind, das ist  $P = P_I$  und das ILP kann durch lineares Programmieren gelöst werden.

## Ein einfaches Lösungsverfahren

Wir wollen  $\max\{c^T x \mid Ax \leq b, x \text{ ganzzahlig}\}$  bestimmen.

Wir starten mit  $\Pi_1 = P, P = \{x \mid Ax \leq b\}$ .

Im Schritt  $k$  haben wir eine Menge  $\Pi_k = \{P_1, \dots, P_k\}$  von Polyedern mit

1.  $P_1, \dots, P_k$  sind paarweise disjunkt (repräsentiert durch lineare Ungleichungen),
2. jeder ganzzahlige Punkt in  $P$  ist in  $P_1 \cup \dots \cup P_k$  enthalten.

Ein Schritt sieht so aus (Eingabe  $\Pi_k$ ):

Sei  $\mu_j = \max\{c^T x \mid x \in P_j\}$  und  $j^*$  so, daß  $\mu_{j^*}$  maximal unter allen  $\mu_j$  ist.

Sei  $x^*$  so, daß  $\mu_{j^*} = c^T x^*$  (diese Werte lassen sich durch lineares Programmieren finden).

Falls  $x^*$  ganzzahlig ist, dann ist  $x^*$  eine optimale Lösung.

Andernfalls, sei  $x_i^*$  eine nicht-ganzzahlige Komponente. Wir definieren

$$Q_1 = \{x \mid P_{j^*} \mid x_i \geq \lceil x_i^* \rceil\}$$

$$Q_2 = \{x \mid P_{j^*} \mid x_i \leq \lfloor x_i^* \rfloor\}$$

Wir setzen

$$\Pi_{k+1} := \{P_1, \dots, P_{j^*-1}, Q_1, Q_2, P_{j^*+1}, \dots, P_k\}.$$

(Falls  $P_i = \emptyset$  für alle  $i$ , dann gibt es keine Lösung.)

Beispiel

