Einführung in Operations Research

Lineare Optimierung (4)

Prof. Dr. Thomas Slawig

CAU Kiel
Institut für Informatik

Inhalt

1. Lineare Optimierung (Lineare Programmierung)
   - Abbruchbedingungen des Simplex-Algorithmus
   - Die zwei Phasen des Simplex-Algorithmus
   - Zyklen im Simplex-Algorithmus
   - Dualität
   - Komplexität des Simplex-Algorithmus

Lernziele

- Die Bedeutung der zwei Phasen des Simplex-Algorithmus verstehen.
- Die Abbruchbedingungen im Simplex-Algorithmus kennen und interpretieren können.
- Die Problematik von Zyklen kennen.
- Den Begriff Dualität und seine Bedeutung verstehen.
- Komplexität des Simplex-Algorithmus kennen.

Inhalt

1. Lineare Optimierung (Lineare Programmierung)
   - Abbruchbedingungen des Simplex-Algorithmus
   - Die zwei Phasen des Simplex-Algorithmus
   - Zyklen im Simplex-Algorithmus
   - Dualität
   - Komplexität des Simplex-Algorithmus

Zusammenfassung: Eine Phase des Simplex-Algorithmus

Input:

- $A \in \mathbb{R}^{m \times n}$, $m < n$, Rang $A = m$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$.
- Basisindizes $\mathcal{B}$, zu denen es zulässige Basislösung gibt.

Algorithmus:

1. Erzeuge Standardbasisvektoren in den Basisspalten und Nullen in der obersten Zeile über den Basisspalten mit Gauß:
$$\begin{array}{c|c} 0 & c^\top \\ \hline b & A \end{array} \rightarrow \begin{array}{c|c} \bar{c}_0 & \bar{c}^\top \\ \hline b' & A' \end{array}$$
2. Wähle $j \notin \mathcal{B}$ mit $\bar{c}_j < 0$, wenn keines existiert: Abbruch.
3. Bestimme ℓ, so dass
$$\frac{b_\ell}{A_{\ell j}} = \min\left\{\frac{b_i}{A_{ij}} : A_{ij} > 0\right\}, \text{ wenn keines existiert: Abbruch.}$$
4. Tausche Basisvariable $\mathcal{B}(\ell) \leftrightarrow j$ und gehe zu 1.

Abbruchbedingung 1: Es gibt kein $j \notin \mathcal{B}$ mit $\bar{c}_j < 0$

Satz

Sei B eine Basis mit Indexmenge $\mathcal{B}$ und $\tilde{A} = B^{-1}A$. Wenn

$$\bar{c}_j = c_j - \sum_{i=1}^{m} c_{\mathcal{B}(i)} \tilde{A}_{ij} \geq 0 \quad \forall j \notin \mathcal{B},$$

dann ist die zu B gehörige Basislösung optimal.

Beweis Optimalität (1)

- Für *jede* Lösung $x \in \mathbb{R}^n$ von $Ax = b$ mit *beliebigen* $x_j, j \notin \mathcal{B}$, gilt:

$$Ax = \sum_{i=1}^{m} x_{\mathcal{B}(i)} A_{*\mathcal{B}(i)} + \sum_{j \notin \mathcal{B}} x_j A_{*j} = b.$$

- Multiplikation von links mit B^{-1}:

$$B^{-1}Ax = \sum_{i=1}^{m} x_{\mathcal{B}(i)} \underbrace{\tilde{A}_{*\mathcal{B}(i)}}_{=e_i} + \sum_{j \notin \mathcal{B}} x_j \tilde{A}_{*j} = B^{-1}b = \tilde{b}.$$

- Auflösen ergibt

$$x_{\mathcal{B}(i)} = \tilde{b}_i - \sum_{j \notin \mathcal{B}} x_j \tilde{A}_{ij}.$$

- $\Rightarrow$ Basisvariablen $x_{\mathcal{B}}$ können durch Nichtbasisvariablen und $\tilde{b}$ ausgedrückt werden.

Beweis Optimalität (2)

- Damit: Kostenfunktion für eine *beliebige* Lösung von $Ax = b$:

$$c^\top x = \sum_{i=1}^{m} c_{\mathcal{B}(i)} \Big(\underbrace{\tilde{b}_i - \sum_{j \notin \mathcal{B}} x_j \tilde{A}_{ij}}_{=x_{\mathcal{B}(i)}} \Big) + \sum_{j \notin \mathcal{B}} c_j x_j = \underbrace{\sum_{i=1}^{m} c_{\mathcal{B}(i)} \tilde{b}_i}_{=c_{\mathcal{B}}^\top x_{\mathcal{B}}} + \sum_{j \notin \mathcal{B}} \Big(\underbrace{c_j - \sum_{i=1}^{m} c_{\mathcal{B}(i)} \tilde{A}_{ij}}_{=:\bar{c}_j} \Big) x_j$$

$$= \begin{cases} c_{\mathcal{B}}^\top x_{\mathcal{B}}, & \text{wenn } x \text{ Basislösung zur Basis } B, (x_{\mathcal{B}} = \tilde{b}, x_j = 0, j \notin \mathcal{B}) \\ c_{\mathcal{B}}^\top x_{\mathcal{B}} + \sum_{j \notin \mathcal{B}} \bar{c}_j x_j, & \text{sonst} \end{cases}$$

- Da diese Darstellung für alle Lösungen von $Ax = b$ gilt:

$$\bar{c}_j \geq 0 \ \ \forall j \notin \mathcal{B} \ \Rightarrow \ \text{Basislösung zu } B \text{ ist optimale Lösung.}$$

□

Abbruchbedingung 2: Für ausgewähltes $j \notin \mathcal{B}$ gilt: $\tilde{A}_{ij} < 0$ für alle i

- Das bedeutet: Es gibt kein ℓ mit
$$\frac{b_\ell}{A_{\ell j}} = \min\left\{\frac{b_i}{A_{ij}} : A_{ij} > 0\right\}.$$
- Dabei sei hier $[b, A]$ die aktuelle erweiterte Matrix im Simplex-Schritt, bei der also die aktuellen Basisspalten schon auf Einheitsvektoren transformiert sind. Dann gilt:
$$Ax = \sum_{i=1}^{m} x_{\mathcal{B}(i)} e_i = b \quad \text{und} \quad \sum_{i=1}^{m} A_{ij} e_i = A_{*j}.$$
- Es gilt für beliebiges $s > 0$:
$$Ax - sA_{*j} + sA_{*j} = \sum_{i=1}^{m} \underbrace{\left(x_{\mathcal{B}(i)} - s\overbrace{A_{ij}}^{<0}\right)}_{>0} e_i + \underbrace{s}_{>0} A_{*j} = \tilde{b}$$
- Auch für $s \to \infty$ bleibt die Gleichung erfüllt: Die zulässige Menge ist unbeschränkt.

Abbruchbedingung 2: Für ausgewähltes $j \notin \mathcal{B}$ gilt: $\tilde{A}_{ij} < 0$ für alle i

- Die zulässige Menge ist unbeschränkt.
- Wir können ℓ beliebig wählen und

$$x'_j = b_\ell - sA_{\ell j}, \quad s > 0 \text{ beliebig groß}$$

  statt $x_{\mathcal{B}(\ell)}$ in die Basis aufnehmen und bleiben zulässig.
- Der Kostenfunktionswert für die neue Basislösung x' ergibt sich dann (s. o. oder VL 2) zu

$$\begin{aligned} c^\top x' = c_\mathcal{B}^\top x_\mathcal{B} + \bar{c}_j x'_j &= c_\mathcal{B}^\top x_\mathcal{B} + \bar{c}_j (b_\ell - sA_{\ell j}) \\ &= c_\mathcal{B}^\top x_\mathcal{B} + \bar{c}_j b_\ell - \underbrace{\underbrace{s}_{>0} \underbrace{\bar{c}_j}_{<0} \underbrace{A_{\ell j}}_{<0}}_{<0} \to -\infty \quad (s \to +\infty). \end{aligned}$$

- Die Kostenfunktion kann beliebig klein werden. Es gibt kein Minimum.

Eine Phase des Simplex-Algorithmus mit Interpretation der Abbruchbedingungen

Input: s.o. Algorithmus:

1. Erzeuge Standardbasisvektoren in den Basisspalten und Nullen in der obersten Zeile über den Basisspalten mit Gauß:
$$\begin{array}{c|c} 0 & c^\top \\ \hline b & A \end{array} \to \begin{array}{c|c} \bar{c}_0 & \bar{c}^\top \\ \hline b' & A' \end{array}$$
2. Wähle $j \notin \mathcal{B}$ mit $\bar{c}_j < 0$, wenn keines existiert: Kostenfunktionswert kann nicht mehr verbessert werden. Aktuelle Basislösung ist optimal $\to$ Abbruch.
3. Bestimme ℓ, so dass
$$\frac{b_\ell}{A_{\ell j}} = \min \left\{ \frac{b_i}{A_{ij}} : A_{ij} > 0 \right\},$$
wenn keines existiert: Kostenfunktion ist nach unten unbeschränkt, es gibt kein Minimum $\to$ Abbruch.
4. Tausche Basisvariable $\mathcal{B}(\ell) \leftrightarrow j$ und gehe zu 1.

Inhalt

1. Lineare Optimierung (Lineare Programmierung)
   - Abbruchbedingungen des Simplex-Algorithmus
   - Die zwei Phasen des Simplex-Algorithmus
   - Zyklen im Simplex-Algorithmus
   - Dualität
   - Komplexität des Simplex-Algorithmus

Zusammenfassung: Zwei Phasen des Simplex-Algorithmus

Input:

- $A \in \mathbb{R}^{m \times n}$, $m < n$, Rang $A = m$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$, zulässige Basislösung nicht bekannt.

Algorithmus:

1. Erreiche $b \geq 0$, ggfs. durch Multiplikation einzelner Zeilen von A mit (-1).
2. Phase 1: Wende eine Phase des Simplex-Algorithmus an auf das Hilfsproblem:

	$x \in \mathbb{R}^n$	$y \in \mathbb{R}^m$
$c_0 := 0$	$0 \cdots 0$	$1 \cdots 1$
b	A	I

   Endergebnis: Transformierte A, b, c_0 und Basislösung.
3. Wenn $c_0 \neq 0$: Zulässige Menge ist leer $\rightarrow$ Abbruch.
4. Wenn $c_0 = 0$: Tausche in Basis alle $y_j = 0$ gegen Nichtbasisvariablen $x_i = 0$.

Zusammenfassung: Zwei Phasen des Simplex-Algorithmus

5. Phase 2: Wende eine Phase des Simplex-Algorithmus an auf das Problem:

$$\begin{array}{c|c} c_0 := 0 & c^\top \\ \hline b & A \end{array}$$

mit den in Phase 1 modifizierten A, b.
Wenn dabei in einem Schritt:

- $\bar{c}_j \geq 0 \; \forall j \notin \mathcal{B}$: Aktuelle Basislösung ist optimal $\rightarrow$ Abbruch.
- $A_{ij} < 0 \; \forall i = 1, \ldots, n$: Kostenfunktion ist nach unten unbeschränkt, es gibt kein Minimum $\rightarrow$ Abbruch.

Output:

- Meldung, wenn zulässige Menge leer oder kein Minimum existiert.
- Sonst: optimale Basislösung x mit $x_{\mathcal{B}} = b$, Wert der Kostenfunktion $c^\top x = -c_0$.

Zusammenfassung: Lösungen eines LP

Für die zulässige Menge eines LP gibt es genau drei Fälle:

- Die zulässige Menge ist leer.
- Die zulässige Menge ist nichtleer und beschränkt.
- Die zulässige Menge ist unbeschränkt.

Für ein LP gibt es genau vier Fälle:

- Die zulässige Menge ist leer, also gibt es keine zulässige und auch keine zulässige optimale Lösung.
- Das Problem hat genau eine zulässige optimale Lösung.
- Das Problem hat unendlich viele zulässige optimale Lösungen, die eine konvexe Menge bilden.
- Die zulässige Menge ist unbeschränkt und die Kostenfunktion auf der zulässigen Menge nach unten unbeschränkt. Also gibt es keine zulässige optimale Lösung.

Inhalt

1. Lineare Optimierung (Lineare Programmierung)
   - Abbruchbedingungen des Simplex-Algorithmus
   - Die zwei Phasen des Simplex-Algorithmus
   - Zyklen im Simplex-Algorithmus
   - Dualität
   - Komplexität des Simplex-Algorithmus

Zyklen

- Zyklen können bei der Basistransformation entstehen, wenn die alte Basislösung entartet/degeneriert war: d. h. es existiert $x_{\mathcal{B}(\ell)} = b_\ell = 0$ mit $A_{\ell j} > 0$.
- Dann wird

$$\min\left\{ \frac{\overbrace{b_i}^{\geq 0}}{A_{ij}} : A_{ij} > 0 \right\} = \frac{\overbrace{b_\ell}^{=0}}{A_{\ell j}} = 0.$$

- Tausch $\mathcal{B}(\ell) \leftrightarrow j$ im Simplex-Schritt bewirkt nichts:

$$b'_\ell := \frac{b_\ell}{A_{\ell j}} = \frac{x_{\mathcal{B}(\ell)}}{A_{\ell j}} = 0, \quad b'_i := b_i - A_{ij} b'_\ell = b_i, \quad i \neq \ell.$$

  $\rightsquigarrow$ neue Basisvariable ebenfalls Null, andere Basisvariablen ändern sich nicht.
- Es wird $b_\ell = x_{\mathcal{B}(\ell)} = 0$ gegen $b'_\ell = x_j = 0$ getauscht.

Beispiel [Papadimitriou Steiglitz]

```
j = argmin{c(j)<0}
l = min{l}
************************************************************************
iter = 0, j = 0, l = 0
   0.00 |  -0.75    20.00    -0.50     6.00     0.00     0.00     0.00
------------------------------------------------------------------------
   0.00  |   0.25    -8.00    -1.00     9.00     1.00     0.00     0.00
   0.00  |   0.50   -12.00    -0.50     3.00     0.00     1.00     0.00
   1.00  |   0.00     0.00     1.00     0.00     0.00     0.00     1.00
B = 5  6  7
************************************************************************
iter = 1, j = 1, l = 1
   0.00 |   0.00    -4.00    -3.50    33.00     3.00     0.00     0.00
------------------------------------------------------------------------
   0.00  |   1.00   -32.00    -4.00    36.00     4.00     0.00     0.00
   0.00  |   0.00     4.00     1.50   -15.00    -2.00     1.00     0.00
   1.00  |   0.00     0.00     1.00     0.00     0.00     0.00     1.00
B = 1  6  7
************************************************************************
iter = 2, j = 2, l = 2
   0.00 |   0.00     0.00    -2.00    18.00     1.00     1.00     0.00
------------------------------------------------------------------------
   0.00  |   1.00     0.00     8.00   -84.00   -12.00     8.00     0.00
   0.00  |   0.00     1.00     0.38    -3.75    -0.50     0.25     0.00
   1.00  |   0.00     0.00     1.00     0.00     0.00     0.00     1.00
B = 1  2  7
```

Beispiel [Papadimitriou Steiglitz]

```
****************************************************************************
iter = 2, j = 2, l = 2
   0.00 |   0.00     0.00    -2.00    18.00     1.00     1.00     0.00
----------------------------------------------------------------------------
   0.00  |   1.00     0.00     8.00   -84.00   -12.00     8.00     0.00
   0.00  |   0.00     1.00     0.38    -3.75    -0.50     0.25     0.00
   1.00  |   0.00     0.00     1.00     0.00     0.00     0.00     1.00
B = 1  2  7
****************************************************************************
iter = 3, j = 3, l = 1
   0.00 |   0.25     0.00     0.00    -3.00    -2.00     3.00     0.00
----------------------------------------------------------------------------
   0.00  |  -0.05     1.00     0.00     0.19     0.06    -0.12     0.00
   0.00  |   0.12     0.00     1.00   -10.50    -1.50     1.00     0.00
   1.00  |  -0.12     0.00     0.00    10.50     1.50    -1.00     1.00
B = 2  3  7
****************************************************************************
iter = 4, j = 4, l = 1
   0.00 |  -0.50    16.00     0.00     0.00    -1.00     1.00     0.00
----------------------------------------------------------------------------
   0.00  |  -2.50    56.00     1.00     0.00     2.00    -6.00     0.00
   0.00  |  -0.25     5.33     0.00     1.00     0.33    -0.67     0.00
   1.00  |   2.50   -56.00     0.00     0.00    -2.00     6.00     1.00
B = 3  4  7
```

Beispiel [Papadimitriou Steiglitz]

```
*****************************************************************************
iter = 4, j = 4, l = 1
   0.00 |  -0.50    16.00     0.00     0.00    -1.00     1.00     0.00
-----------------------------------------------------------------------------
   0.00  |  -2.50    56.00     1.00     0.00     2.00    -6.00     0.00
   0.00  |  -0.25     5.33     0.00     1.00     0.33    -0.67     0.00
   1.00  |   2.50   -56.00     0.00     0.00    -2.00     6.00     1.00
B = 3  4  7
*****************************************************************************
iter = 5, j = 5, l = 1
   0.00 |  -1.75    44.00     0.50     0.00     0.00    -2.00     0.00
-----------------------------------------------------------------------------
   0.00  |   0.17    -4.00    -0.17     1.00     0.00     0.33     0.00
   0.00  |  -1.25    28.00     0.50     0.00     1.00    -3.00     0.00
   1.00  |   0.00     0.00     1.00     0.00     0.00     0.00     1.00
B = 4  5  7
*****************************************************************************
iter = 6, j = 6, l = 1
   0.00 |  -0.75    20.00    -0.50     6.00     0.00     0.00     0.00
-----------------------------------------------------------------------------
   0.00  |   0.25    -8.00    -1.00     9.00     1.00     0.00     0.00
   0.00  |   0.50   -12.00    -0.50     3.00     0.00     1.00     0.00
   1.00  |   0.00     0.00     1.00     0.00     0.00     0.00     1.00
B = 5  6  7
```

Methoden zur Vermeidung von Zyklen

Methode von Bland (war in VL 2 schon so angegeben):

- Wähle nicht $j \notin \mathcal{B}$ mit kleinstem Wert $\bar{c}_j < 0$ (größte Reduktion der Kostenfunktion),
- ... sondern *kleinsten (ersten) Index* $j \notin \mathcal{B}$ mit $\bar{c}_j < 0$.
- ... und ℓ so, dass $\mathcal{B}(\ell)$ am kleinsten ist, wenn es nach der Regel

$$\min\left\{\frac{b_i}{A_{ij}} : A_{ij} > 0\right\} = \frac{b_\ell}{A_{\ell j}}$$

  mehrere Indizes ℓ gibt, wo das Minimum angenommen wird.

Lexikographische Regel:

- Wähle *beliebiges* $j \notin \mathcal{B}$ mit $\bar{c}_j < 0$ (z. B. das mit minimalem Wert).
- Betrachte die *gesamten Zeilen* des Tableaus mit $A_{ij} > 0$ und wähle ℓ so, dass

$$\frac{1}{A_{\ell j}}[b_i, A_{\ell *}^\top] <_L \frac{1}{A_{ij}}[b_i, A_{i*}^\top] \quad \forall i = 1, \ldots, m, A_{ij} > 0.$$

  wobei für $v <_L w$ die *erste Komponente* entscheidet, die kleiner ist.

Inhalt

1. Lineare Optimierung (Lineare Programmierung)
   - Abbruchbedingungen des Simplex-Algorithmus
   - Die zwei Phasen des Simplex-Algorithmus
   - Zyklen im Simplex-Algorithmus
   - Dualität
   - Komplexität des Simplex-Algorithmus

Dualität: Begriff

- Zu einem LP in Standardform (auch **primales** LP)

$$\min_{x \in \mathbb{R}^n} c^\top x, \quad Ax = b, x \geq 0, \quad A \in \mathbb{R}^{m \times n}$$

- definieren wir das dazu **duale** LP als

$$\max_{y \in \mathbb{R}^m} b^\top y, \quad A^\top y \leq c \quad \left(\Leftrightarrow y^\top A \leq c^\top\right).$$

- Beziehung primales $\leftrightarrow$ duales LP:
  Kostenvektor c und b tauschen Rollen, Minimum wird zu Maximum.
- Es gilt: Duales LP des dualen LP = primales LP.
  (Transformiere duales LP in Standardform ...)

Beispiel

- Primales LP:

$$\min_{x\in\mathbb{R}^3} (-1x_1 - 4x_2 - 3x_3) \text{ bei } \begin{cases} 2x_1 + 2x_2 + x_3 = 4 \\ x_1 + 2x_2 + 2x_3 = 6 \\ x \geq 0 \end{cases}$$

- Duales LP:

$$\max_{y\in\mathbb{R}^2} 4y_1 + 6y_2 \text{ bei } \begin{cases} 2y_1 + y_2 \leq -1 \\ 2y_1 + 2y_2 \leq -4 \\ y_1 + 2y_2 \leq -3 \end{cases}$$

Dualität: Beziehungen

Lemma (Dualitätslemma)

Für jedes zulässige Paar (x, y) *des primalen bzw. dualen Problems gilt*

$$b^\top y = y^\top b = \underbrace{y^\top A}_{\leq c^\top} \underbrace{x}_{\geq 0} \leq c^\top x,$$

d. h. die Kostenfunktion des dualen Problems ist immer kleiner oder gleich der des primalen.

Folgerung

Sind x, y *wie oben jeweils zulässig mit* $c^\top x = b^\top y$*, dann sind* x *und* y *optimal.*

Dualität: Beziehungen

Satz (Dualitätssatz der LP)

- *Hat das primale oder das duale LP eine optimale Lösung, dann auch das andere. Beide Kostenfunktionswerte sind gleich.*
- *Hat das primale LP eine optimale zulässige Lösung zur Basis B, dann ist $y \in \mathbb{R}^m$ mit $y^\top = c_B^\top B^{-1}$ eine optimale Lösung des dualen LP.*
- *Hat eines der beiden Probleme eine auf der jeweiligen zulässigen Menge unbeschränkte Kostenfunktion, dann hat das andere keine zulässige Lösung.*

Duale Lösung im Simplex-Tableau

- Dualitätssatz: x (Basis B) optimal für primales LP $\Rightarrow y^\top = c_B^\top B^{-1}$ optimal für duales LP.
- Ergebnis nach Phase 1 (in der Zeichnung: Basisspalten vorne):

$$A \to [\ \underbrace{\cdots I \cdots}_{\text{Basisspalten}},\ \underbrace{\cdots \tilde{A} \cdots}_{\text{Nichtbasisspalten}}\ ].$$

- Ergebnis nach Phase 2:

$$[\ \underbrace{\cdots I \cdots}_{\text{Basisspalten}},\ \underbrace{\cdots \tilde{A} \cdots}_{\text{Nichtbasisspalten}}\ ] \to B^{-1}[I, \tilde{A}] = [\ \underbrace{\cdots B^{-1} \cdots}_{\text{alte Basisspalten}},\ \underbrace{\cdots I \cdots}_{\text{neue Basisspalten}}\ ].$$

- B^{-1} steht da, wo zu Beginn von Phase 2 die Basisspalten waren.

Duale Variable y als Maß der Sensitivität

- Sei $(x_B, 0) = (B^{-1}b, 0)$ die optimale Lösung des primalen LP mit Basis B.
- Dualitätssatz $\Rightarrow y^\top = c_{\mathcal{B}}^\top B^{-1}$ ist Lösung des dualen LP.
- Annahme: Nichtdegeneriertheit der Lösung
- Betrachte kleine Störung Δb in b (so, dass sich B nicht ändert).
- Optimale Lösung für die gestörte Nebenbedingung $Ax = b + \Delta b$:

$$x = (x_{\mathcal{B}} + \Delta x_{\mathcal{B}}, 0) \quad \text{mit} \quad \Delta x_{\mathcal{B}} = B^{-1}\Delta b.$$

- Die Änderung in der Kostenfunktion ist dann

$$\Delta z = c_{\mathcal{B}}^\top \Delta x_{\mathcal{B}} = c_{\mathcal{B}}^\top B^{-1}\Delta b = y^\top \Delta b.$$

- An den Komponenten von y ist also zu erkennen, wie sich eine Störung Δb der entsprechenden Komponente der Nebenbedingung auf die Kostenfunktion auswirkt.

Komplementarität

- Seien x, y zulässig für das primale bzw. duale Problem:

$$\min_{x \in \mathbb{R}^n} c^\top x,\ Ax = b,\ x \geq 0, \qquad \max_{y \in \mathbb{R}^m} y^\top b,\ y^\top A \leq c^\top.$$

- Dualitätssatz: x, y sind genau dann jeweils optimal, wenn

$$c^\top x = y^\top b = y^\top Ax \iff (y^\top A - c^\top)x = 0$$
$$\iff \sum_{i=1}^{n} \underbrace{(y^\top A_{*i} - c_i)}_{\leq 0} \underbrace{x_i}_{\geq 0} = 0$$

- Notwendig und hinreichend für die Optimalität von x und y ist also:

$$\left\{ \begin{array}{rclcrcl} x_i & > & 0 & \Rightarrow & y^\top A_{*i} & = & c_i \\ y^\top A_{*i} & < & c_i & \Rightarrow & x_i & = & 0 \end{array} \right\} \quad \forall i = 1, \ldots, n.$$

- Also ist komponentenweise jeweils eine Ungleichheitsnebenbedingung des primalen oder dualen Problems **aktiv** (d. h. es gilt Gleichheit).

Inhalt

1. Lineare Optimierung (Lineare Programmierung)
   - Abbruchbedingungen des Simplex-Algorithmus
   - Die zwei Phasen des Simplex-Algorithmus
   - Zyklen im Simplex-Algorithmus
   - Dualität
   - Komplexität des Simplex-Algorithmus

Simplex ist nicht polynomial im Aufwand

- $A \in \mathbb{R}^{m\times n}$: „meistens“: $\#$ Iterationen $\approx 3m$
- Beweisversuche: Simplex ist polynomial im Aufwand
- Beispiel (Klee, Minty):

$$\max_{x\in\mathbb{R}^n} \sum_{j=1}^{n} 10^{n-j} x_j \text{ bei } \begin{cases} 2\sum_{j=1}^{i-1} 10^{i-j} x_j + x_i \le 100^{i-1}, i = 1, \dots, n \\ x \ge 0 \end{cases}$$

$$n = 3: \quad \max_{x\in\mathbb{R}^3} 100x_1 + 10x_2 + x_3 \text{ bei } \begin{cases} x_1 & \le 1 \\ 20x_1 + x_2 & \le 100 \\ 200x_1 + 20x_2 + x_3 & \le 10000 \\ x & \ge 0 \end{cases}$$

in Standardform: $m = 3, n = 6$: Simplex braucht $2^3 - 1 = 7$ Schritte

- $m \in \mathbb{N}$: $2^m - 1$ Schritte $=$ alle Ecken des Polytops außer Startecke.
- $m = 50 : 2^{50} \approx 10^{15}$: bei 10^6 Schritten/s $\rightsquigarrow$ 33 Jahre.

Lernziele

- Die Bedeutung der zwei Phasen des Simplex-Algorithmus verstehen.
- Die Abbruchbedingungen im Simplex-Algorithmus kennen und interpretieren können.
- Die Problematik von Zyklen kennen.
- Den Begriff Dualität und seine Bedeutung verstehen.
- Komplexität des Simplex-Algorithmus kennen.