Technische Universität Braunschweig

Mathematische Methoden der Algorithmik – Vorlesung #05

Arne Schmidt

Letzte Woche

Dualer Simplex-Algorithmus

Dualer Simplex-Algorithmus

$$\begin{array}{rcrrr}
\zeta & = & 0 - & 1x_1 - & 1x_2 \\
\hline
w_1 & = & 4 + & 2x_1 + & 1x_2 \\
w_2 & = & \boxed{-8} + & \boxed{2x_1} - & 4x_2 \\
w_3 & = & -7 + & 1x_1 - & 3x_2
\end{array}$$

$$\begin{array}{rcrrr}
\zeta & = & 0 - & 1x_1 + & 4x_2 \\
\hline
w_1 & = & 4 + & 2x_1 + & 1x_2 \\
w_2 & = & -8 + & 2x_1 - & 4x_2 \\
w_3 & = & -7 + & 1x_1 - & 3x_2
\end{array}$$

$\downarrow$ Duale Phase I

$$\begin{array}{rcrrr}
\eta & = & 0 - & 1x_1 - & 1x_2 \\
\hline
w_1 & = & 4 + & 2x_1 + & 1x_2 \\
w_2 & = & -8 + & 2x_1 - & 4x_2 \\
w_3 & = & -7 + & 1x_1 - & 3x_2
\end{array}$$

LPs in Matrix-Notation

LPs in Matrixnotation

Betrachte ein LP in Standardform, für welches Schlupfvariablen $x_{n+1}, \dots, x_{n+m}$ eingeführt werden. Dann haben wir die Form:

$$\begin{aligned} \max \quad & c^T x \\ s.t. \quad & Ax = b \\ & x \geq 0 \end{aligned}$$

Wobei

$$A = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} & 1 & 0 & \cdots & 0 \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} & 0 & 0 & \cdots & 1 \end{bmatrix}, \quad b = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}, \quad c = \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \quad x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \\ x_{n+1} \\ \vdots \\ x_{n+m} \end{pmatrix}$$

Basis- und Nichtbasisteile

$$A = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} & 1 & 0 & \cdots & 0 \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} & 0 & 0 & \cdots & 1 \end{bmatrix}, \quad b = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}, \quad c = \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \quad x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \\ x_{n+1} \\ \vdots \\ x_{n+m} \end{pmatrix}$$

Im Simplex benutzen wir Basis und Nichtbasisvariablen.
Dann gilt für die i-te Komponente von Ax:

$$\sum_{j=1}^{n+m} a_{ij} x_j = \sum_{j \in \mathcal{B}} a_{ij} x_j + \sum_{j \in \mathcal{N}} a_{ij} x_j$$

Oder in Matrixschreibweise:

$$A = [B \;\; N], x = \begin{bmatrix} x_{\mathcal{B}} \\ x_{\mathcal{N}} \end{bmatrix}$$

B ist eine $m \times m$-Matrix mit Spalten aus A, die zu Indizes aus $\mathcal{B}$ gehören.

N ist eine $m \times n$-Matrix mit Spalten aus A, die zu Indizes aus $\mathcal{N}$ gehören.

Summe von Basis- und Nichtbasisteile

Also ist

$$Ax = [B \;\; N] \begin{bmatrix} x_{\mathcal{B}} \\ x_{\mathcal{N}} \end{bmatrix} = Bx_{\mathcal{B}} + Nx_{\mathcal{N}}$$

Sowie

$$c^T x = [c_{\mathcal{B}} \;\; c_{\mathcal{N}}] \begin{bmatrix} x_{\mathcal{B}} \\ x_{\mathcal{N}} \end{bmatrix} = c_{\mathcal{B}}^T x_{\mathcal{B}} + c_{\mathcal{N}}^T x_{\mathcal{N}}$$

Beispiel: Schreibe das folgende LP in Matrixform und bestimme die Werte für $\mathcal{B} = \{1,2\}$.

$$\begin{array}{rrrrl} \max\limits_{x} & 3x_1 + & 4x_2 - & 2x_3 & \\ \text{subject to} & x_1 + & 0.5x_2 - & 5x_3 & \leq 2 \\ & 2x_1 - & x_2 + & 3x_3 & \leq 3 \\ & x_1, & x_2, & x_3 & \geq 0 \end{array}$$

Simplex in Matrix-Notation
–
Primaler Simplex-Algorithmus

Basisvariablen Matrixform

In Dictionaries haben wir die Basisvariablen als Funktion von Nichtbasisvariablen betrachtet:

$$x_i = \bar{b}_i - \sum_{j \in \mathcal{N}} \bar{a}_{ij} x_j$$

Aus der Matrixform $Ax = b$ bzw. $Bx_{\mathcal{B}} + Nx_{\mathcal{N}} = b$ können wir $x_{\mathcal{B}}$ definieren als

$$x_{\mathcal{B}} = B^{-1}b \ - B^{-1}Nx_{\mathcal{N}}$$

Die Matrix B muss also invertierbar sein, d.h. sie muss
- Quadratisch sein, sowie
- vollen Rang besitzen (rg(B) = m), d.h. m linear unabhängige Spaltenvektoren enthalten.

Wir haben also eine Basis für $\mathbb{R}^m$!

Zielfunktion in Matrixform

Damit können wir auch die Zielfunktion aus dem Dictionary in Matrixform angeben:

$$\begin{aligned}\zeta &= c_{\mathcal{B}}^T x_{\mathcal{B}} + c_{\mathcal{N}}^T x_{\mathcal{N}} \\ &= c_{\mathcal{B}}^T (B^{-1} b - B^{-1} N x_{\mathcal{N}}) + c_{\mathcal{N}}^T x_{\mathcal{N}} \\ &= c_{\mathcal{B}}^T B^{-1} b - \left((B^{-1} N)^T c_{\mathcal{B}} - c_{\mathcal{N}}\right)^T x_{\mathcal{N}}\end{aligned}$$

Damit haben wir folgende Matrixform für Dictionaries:

$$\begin{aligned}\zeta &= c_{\mathcal{B}}^T B^{-1} b - \left((B^{-1} N)^T c_{\mathcal{B}} - c_{\mathcal{N}}\right)^T x_{\mathcal{N}} \\ x_{\mathcal{B}} &= B^{-1} b - B^{-1} N x_{\mathcal{N}}\end{aligned}$$

Dictionaries in Matrixform

Damit haben wir folgende Matrixform für Dictionaries:

$$\begin{aligned} \zeta &= \boxed{c_{\mathcal{B}}^T B^{-1} b} \boxed{- \left((B^{-1}N)^T c_{\mathcal{B}} - c_{\mathcal{N}} \right)^T x_{\mathcal{N}}} \\ x_{\mathcal{B}} &= \boxed{B^{-1} b} - \boxed{B^{-1} N x_{\mathcal{N}}} \end{aligned}$$

In alter Darstellung:

$$\zeta = \boxed{\bar{\zeta}} + \boxed{\sum_{j \in \mathcal{N}}^{n} \bar{c}_j x_j}$$

$$x_i = \boxed{\bar{b}_i} - \boxed{\sum_{j=1}^{n} \bar{a}_{ij} x_j}, \qquad i \in \mathcal{B}$$

Also:

$$\begin{aligned} \bar{\zeta} &= c_{\mathcal{B}}^T B^{-1} b \\ [\bar{c}_j] &= c_{\mathcal{N}} - (B^{-1}N)^T c_{\mathcal{B}} \\ [\bar{b}_i] &= B^{-1} b \\ [\bar{a}_{ij}] &= -B^{-1} N \end{aligned}$$

Die eckigen Klammern drücken dabei den Vektor bzw. die Matrix über die Laufvariablen aus.

Basislösungen

Da die Nichtbasisvariablen auf 0 gesetzt werden, sind Lösungen im Dictionary also vereinfacht:

$$x_{\mathcal{N}}^* = 0, x_{\mathcal{B}}^* = B^{-1}b$$

Also braucht man für eine gegebene Basis $\mathcal{B}$ nur das lineare Gleichungssystem $Bx_{\mathcal{B}} = b$ lösen. Alle anderen Variablen werden auf 0 gesetzt.

Allerdings kann für eine Basis $\mathcal{B}$ auch eine ungültige Lösung entstehen: $x_{\mathcal{B}}^*$ können so negativ sein. Wir können $\mathcal{B}$ also nicht einfach raten.

Am Beispiel: Wir haben die Basis $\mathcal{B} = \{2, 5\}$.
Bestimme die anderen Teile des Dictionaries.

$$\begin{array}{lrrrl}
\max\limits_{x} & 3x_1 + & 4x_2 - & 2x_3 & \\
\text{subject to} & x_1 + & 0.5x_2 - & 5x_3 & \leq 2 \\
 & 2x_1 - & x_2 + & 3x_3 & \leq 3 \\
 & x_1, & x_2, & x_3 & \geq 0
\end{array}$$

Dualer Simplex – Vorüberlegungen

Für das duale Dictionary haben wir gesehen, dass die Wertematrix A im dualen negativ transponiert ist. Es ist also wichtig, die komplementären Paare von Variablen zu kennen.

Primal: $(x_1, \dots, x_n, w_1, \dots, w_m) \rightarrow (x_1, \dots, x_n, x_{n+1}, \dots, x_{n+m})$

Dual: $(z_1, \dots, z_n, y_1, \dots, y_m) \rightarrow (z_1, \dots, z_n, z_{n+1}, \dots, z_{n+m})$

Duales Dictionary

Für das primale Dictionary haben wir:

$$\zeta = c_{\mathcal{B}}^T B^{-1} b - \left((B^{-1}N)^T c_{\mathcal{B}} - c_{\mathcal{N}}\right)^T x_{\mathcal{N}}$$
$$x_{\mathcal{B}} = B^{-1} b - B^{-1} N x_{\mathcal{N}}$$

Lösung primal:

$$x_{\mathcal{B}}^* = B^{-1} b,$$
$$x_{\mathcal{N}}^* = 0$$

Dictionary in Kurzform

$$\zeta = \zeta^* - (z_{\mathcal{N}}^*)^T x_{\mathcal{N}}$$
$$x_{\mathcal{B}} = x_{\mathcal{B}}^* - B^{-1} N x_{\mathcal{N}}$$

Dann können wir für das duale Dictionary folgendes schreiben

$$-\xi = -c_{\mathcal{B}}^T B^{-1} b \qquad - (B^{-1} b)^T z_{\mathcal{B}}$$
$$z_{\mathcal{N}} = \left((B^{-1}N)^T c_{\mathcal{B}} - c_{\mathcal{N}}\right) + (B^{-1}N)^T z_{\mathcal{B}}$$

Lösung dual:

$$z_{\mathcal{B}}^* = 0,$$
$$z_{\mathcal{N}}^* = \left((B^{-1}N)^T c_{\mathcal{B}} - c_{\mathcal{N}}\right)$$

Dictionary in Kurzform

$$-\xi = -\zeta^* - (x_{\mathcal{B}}^*)^T z_{\mathcal{B}}$$
$$z_{\mathcal{N}} = z_{\mathcal{N}}^* + (B^{-1}N)^T z_{\mathcal{B}}$$

Primaler Simplex in Matrixform

Primal:

$$\zeta = \zeta^* \quad - (z_{\mathcal{N}}^*)^T x_{\mathcal{N}}$$
$$x_{\mathcal{B}} = x_{\mathcal{B}}^* - B^{-1} N x_{\mathcal{N}}$$

Dual:

$$-\xi = -\zeta^* - (x_{\mathcal{B}}^*)^T z_{\mathcal{B}}$$
$$z_{\mathcal{N}} = z_{\mathcal{N}}^* \quad + (B^{-1}N)^T z_{\mathcal{B}}$$

Annahme: Basis $\mathcal{B}$ gegeben mit

- m Elementen
- B ist invertierbar
- $x_{\mathcal{B}}^* = B^{-1} b \geq 0$

Pivotvorbereitung:

1. Falls $z_{\mathcal{N}}^* \geq 0$, dann stoppe (Optimalität)
2. Ansonsten, wähle $j \in \mathcal{N}$ mit $z_j^* < 0$. (Variable mit möglicher Verbesserung)
3. Bestimme $\Delta x_{\mathcal{B}} = B^{-1} N e_j$, wobei e_j ein Null-Vektor mit einer 1 an der j-ten Position ist.
4. Bestimme größtes $t \geq 0$, sodass $x_{\mathcal{B}}^* \geq t \Delta x_{\mathcal{B}}$.
   a. Also: $t = \left(\max_{i \in \mathcal{B}} \frac{\Delta x_{\mathrm{i}}}{x_i^*} \right)^{-1}$ (0/0 := 0)
   b. Ist $t \leq 0$, stoppe (Unbeschränkt)
5. Wähle $i \in \mathcal{B}$, über welche t definiert wird.

Primaler Simplex in Matrixform

Primal:

$\zeta = \zeta^* \quad - (z_{\mathcal{N}}^*)^T x_{\mathcal{N}}$

$x_{\mathcal{B}} = x_{\mathcal{B}}^* - B^{-1} N x_{\mathcal{N}}$

Dual:

$-\xi = -\zeta^* - (x_{\mathcal{B}}^*)^T z_{\mathcal{B}}$

$z_{\mathcal{N}} = z_{\mathcal{N}}^* \quad + (B^{-1}N)^T z_{\mathcal{B}}$

Annahme: Basis $\mathcal{B}$ gegeben mit

- m Elementen
- B ist invertierbar
- $x_{\mathcal{B}}^* = B^{-1} b \geq 0$

Pivotschritt (tausche $\boldsymbol{j \in \mathcal{N}}$ mit $\boldsymbol{i \in \mathcal{B}}$):

1. Bestimme $\Delta z_{\mathcal{N}} = -(B^{-1}N)^T e_i$
2. Berechne $s = \dfrac{z_j^*}{\Delta z_j}$
3. Aktualisiere primal und duale Lösungen:
   a. $x_j^* := t$ und $x_{\mathcal{B}}^* := x_{\mathcal{B}}^* - t \Delta x_{\mathcal{B}}$
   b. $z_i^* := s$ und $z_{\mathcal{N}}^* := z_{\mathcal{N}}^* - s \Delta z_{\mathcal{N}}$
4. Setze $\mathcal{B} := (\mathcal{B} \setminus \{i\}) \cup \{j\}$ (Basis aktualisieren)

Ein Beispiel

$$
\begin{aligned}
\max_{x} \quad & 4x_1 + 3x_2 \\
\text{subject to} \quad & x_1 - x_2 \leq 1 \\
& 2x_1 - x_2 \leq 3 \\
& + x_2 \leq 5 \\
& x_1, \; x_2 \geq 0
\end{aligned}
$$

Simplex in Matrix-Notation

–

Dualer Simplex-Algorithmus

Dualer Simplex in Matrixform

Primal:

$$\zeta = \zeta^* \quad - (z_{\mathcal{N}}^*)^T x_{\mathcal{N}}$$
$$x_{\mathcal{B}} = x_{\mathcal{B}}^* - B^{-1} N x_{\mathcal{N}}$$

Dual:

$$-\xi = -\zeta^* - (x_{\mathcal{B}}^*)^T z_{\mathcal{B}}$$
$$z_{\mathcal{N}} = z_{\mathcal{N}}^* \quad + (B^{-1}N)^T z_{\mathcal{B}}$$

Annahme: Basis $\mathcal{B}$ gegeben mit

- m Elementen
- B ist invertierbar
- $z_{\mathcal{N}}^* = \left((B^{-1}N)^T c_{\mathcal{B}} - c_{\mathcal{N}}\right) \geq 0$

Pivotvorbereitung:

1. Falls $x_{\mathcal{B}}^* \geq 0$, dann stoppe (Optimalität)
2. Ansonsten, wähle $i \in \mathcal{B}$ mit $x_i^* < 0$. (Variable mit möglicher Verbesserung)
3. Bestimme $\Delta z_{\mathcal{B}} = -(B^{-1}N)^T e_j$
4. Bestimme Pivotschrittlänge.
   a. $s = \left(\max_{j \in \mathcal{N}} \frac{\Delta \mathrm{z_j}}{z_j^*} \right)^{-1}$ $(0/0 := 0)$
   b. Ist $s \leq 0$, stoppe (Dual unbeschränkt, primal infeasbile)
5. Wähle $j \in \mathcal{N}$, über welche s definiert wird.

Dualer Simplex in Matrixform

Primal:
$$\zeta = \zeta^* \quad - (z_{\mathcal{N}}^*)^T x_{\mathcal{N}}$$
$$x_{\mathcal{B}} = x_{\mathcal{B}}^* - B^{-1} N x_{\mathcal{N}}$$

Dual:
$$-\xi = -\zeta^* - (x_{\mathcal{B}}^*)^T z_{\mathcal{B}}$$
$$z_{\mathcal{N}} = z_{\mathcal{N}}^* \quad + (B^{-1}N)^T z_{\mathcal{B}}$$

Annahme: Basis $\mathcal{B}$ gegeben mit
- m Elementen
- B ist invertierbar
- $z_{\mathcal{N}}^* = \left((B^{-1}N)^T c_{\mathcal{B}} - c_{\mathcal{N}}\right) \geq 0$

Dualer Pivotschritt (tausche $\boldsymbol{j} \in \boldsymbol{\mathcal{N}}$ mit $\boldsymbol{i} \in \boldsymbol{\mathcal{B}}$):

1. Bestimme $\Delta x_{\mathcal{B}} = B^{-1} N e_j$
2. Berechne $t = \dfrac{x_i^*}{\Delta x_i}$
3. Aktualisiere primale und duale Lösungen:
   a. $z_i^* := s$ und $z_{\mathcal{N}}^* := z_{\mathcal{N}}^* - s\Delta z_{\mathcal{N}}$
   b. $x_j^* := t$ und $x_{\mathcal{B}}^* := x_{\mathcal{B}}^* - t\Delta x_{\mathcal{B}}$
4. Setze $\mathcal{B} := (\mathcal{B} \setminus \{i\}) \cup \{j\}$ (Basis aktualisieren)

Simplex in Matrixform

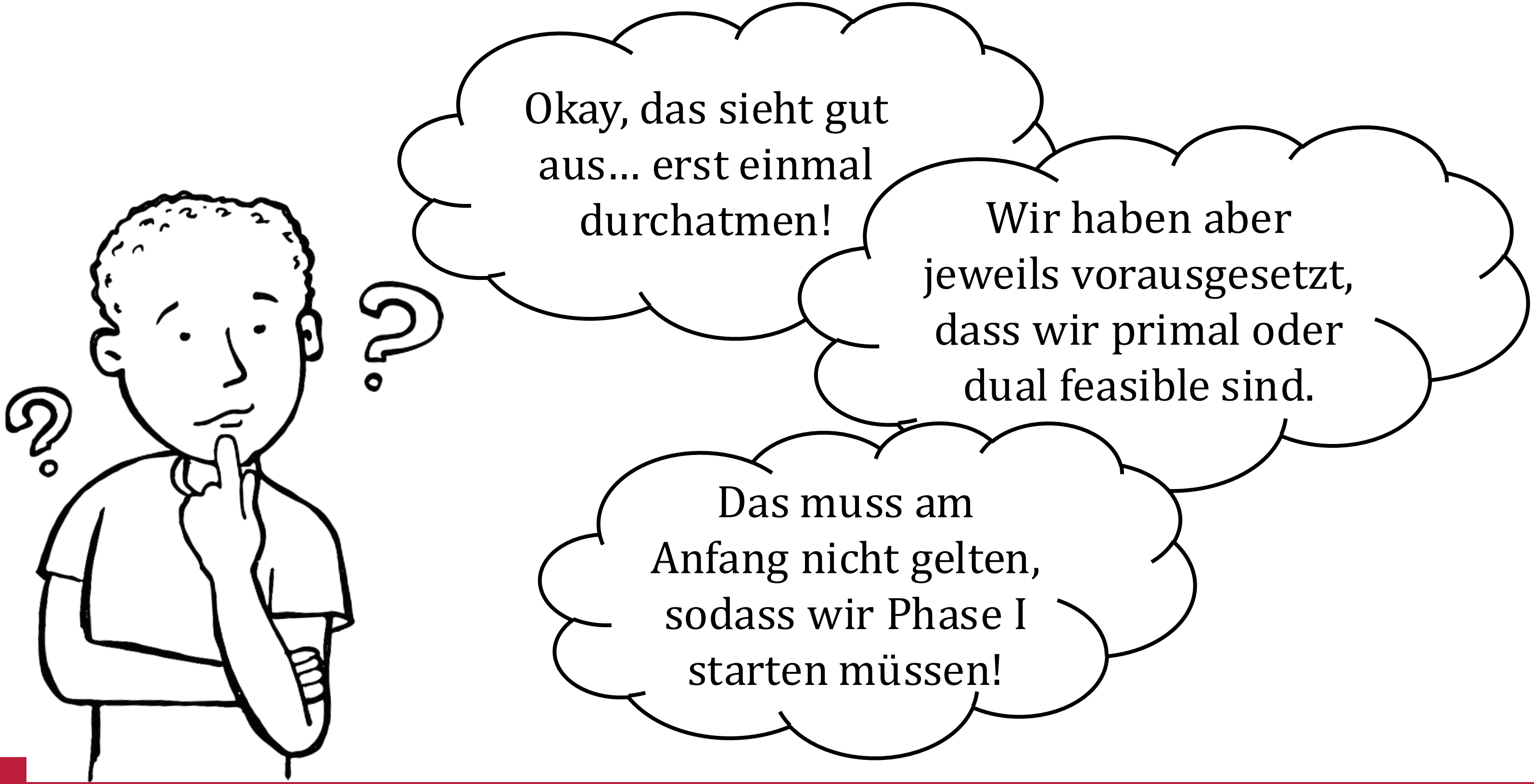
Primal Simplex

Suppose $x^*_{\mathcal{B}} \geq 0$
while $(z^*_{\mathcal{N}} \not\geq 0)$ {
  pick $j \in \{j \in \mathcal{N} : z^*_j < 0\}$
  $\Delta x_{\mathcal{B}} = B^{-1} N e_j$
  $t = \left(\max_{i \in \mathcal{B}} \frac{\Delta x_i}{x^*_i} \right)^{-1}$
  pick $i \in \operatorname{argmax}_{i \in \mathcal{B}} \frac{\Delta x_i}{x^*_i}$
  $\Delta z_{\mathcal{N}} = -(B^{-1} N)^T e_i$
  $s = \frac{z^*_j}{\Delta z_j}$
  $x^*_j \leftarrow t$
  $x^*_{\mathcal{B}} \leftarrow x^*_{\mathcal{B}} - t \Delta x_{\mathcal{B}}$
  $z^*_i \leftarrow s$
  $z^*_{\mathcal{N}} \leftarrow z^*_{\mathcal{N}} - s \Delta z_{\mathcal{N}}$
  $\mathcal{B} \leftarrow \mathcal{B} \setminus \{i\} \cup \{j\}$
}

Dual Simplex

Suppose $z^*_{\mathcal{N}} \geq 0$
while $(x^*_{\mathcal{B}} \not\geq 0)$ {
  pick $i \in \{i \in \mathcal{B} : x^*_i < 0\}$
  $\Delta z_{\mathcal{N}} = -(B^{-1} N)^T e_i$
  $s = \left(\max_{j \in \mathcal{N}} \frac{\Delta z_j}{z^*_j} \right)^{-1}$
  pick $j \in \operatorname{argmax}_{j \in \mathcal{N}} \frac{\Delta z_j}{z^*_j}$
  $\Delta x_{\mathcal{B}} = B^{-1} N e_j$
  $t = \frac{x^*_i}{\Delta x_i}$
  $x^*_j \leftarrow t$
  $x^*_{\mathcal{B}} \leftarrow x^*_{\mathcal{B}} - t \Delta x_{\mathcal{B}}$
  $z^*_i \leftarrow s$
  $z^*_{\mathcal{N}} \leftarrow z^*_{\mathcal{N}} - s \Delta z_{\mathcal{N}}$
  $\mathcal{B} \leftarrow \mathcal{B} \setminus \{i\} \cup \{j\}$
}

Kurze Denkpause

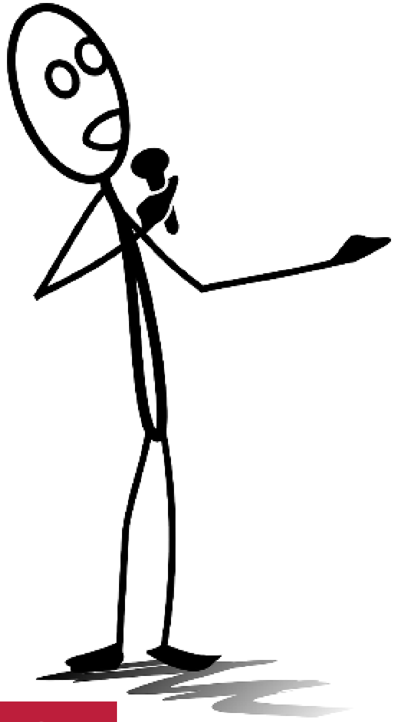


Simplex in Matrix-Notation

–

Zwei-Phasen Methode

Situation am Anfang

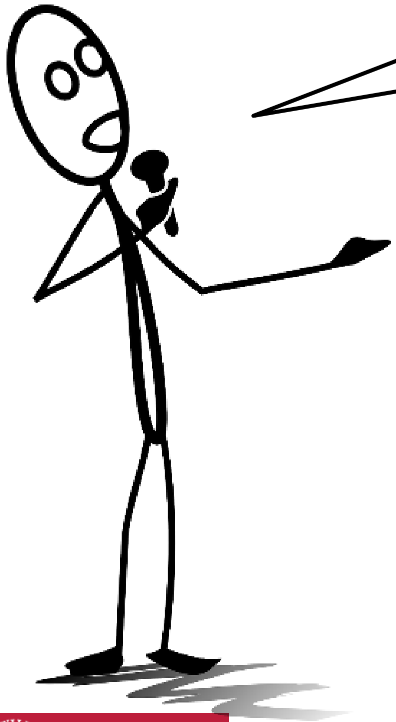


Initial, haben wir folgendes Dictionary

$$\boxed{\begin{aligned} \zeta &= \quad c_{\mathcal{N}}^T x_{\mathcal{N}} \\ x_{\mathcal{B}} &= b - N x_{\mathcal{N}} \end{aligned}}$$

1. $b \geq 0$ und $c \leq 0$ → Primal feasible und optimal!
2. $b \geq 0$ und $c \nleq 0$ → Primal feasible, starte primal Simplex
3. $b \ngeq 0$ und $c \leq 0$ → Dual Feasible, starte dual Simplex
4. $b \ngeq 0$ und $c \nleq 0$ → Starte Phase I

Möglichkeiten für Phase I



Für die Phase I stehen 3 Möglichkeiten zur Verfügung!

Benutze das Hilfsproblem aus Vorlesung 2 und führe primalen Simplex aus.

Ersetze c durch einen nicht-positiven Vektor und führe dualen Simplex aus.

Phase II: Primaler Simplex

Ersetze b durch einen nicht-negativen Vektor und führe primalen Simplex aus.

Phase II: Dualer Simplex

Nächste Woche

Damit haben wir einen Pseudocode, der einfach umsetzbar ist.

Was müssen wir bei einer Implementierung beachten?

Können wir irgendwo Arbeit sparen?