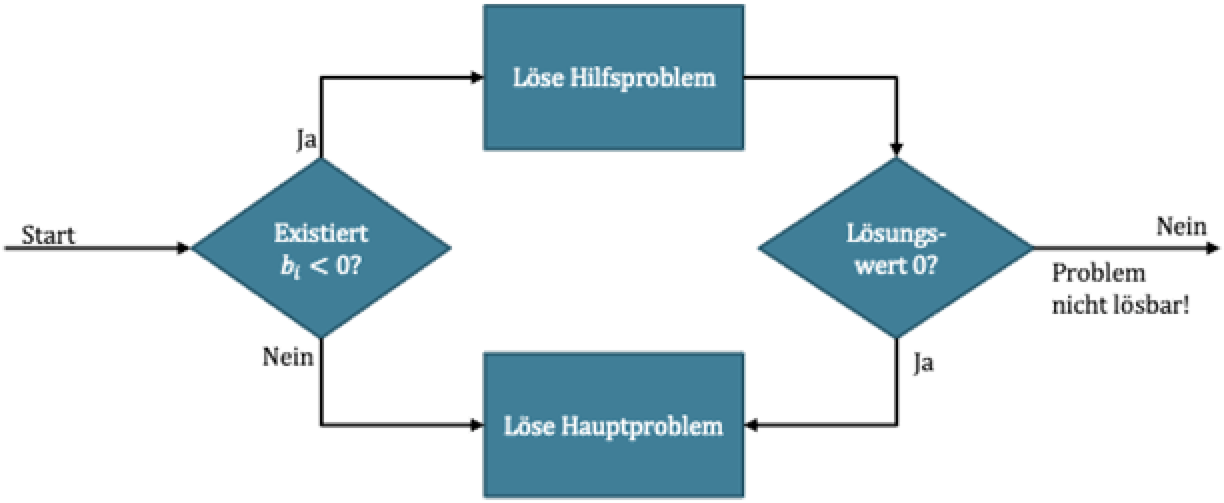
Mathematatische Methoden der Algorithmik – Vorlesung #02

Arne Schmidt

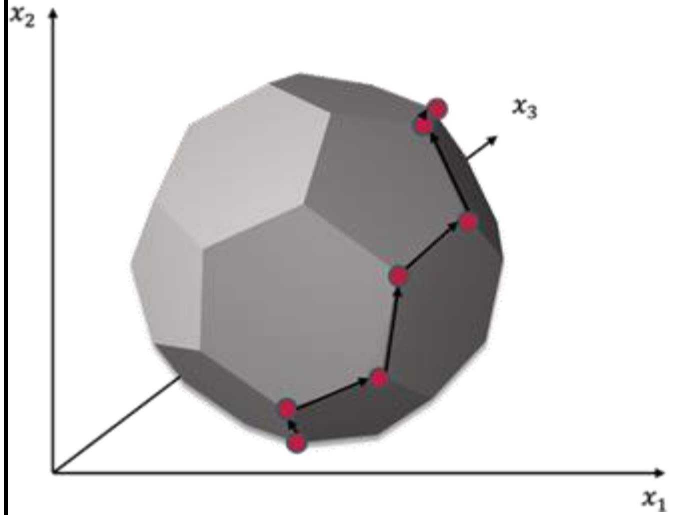
Letzte Woche

Simplex Algorithmus

2-Phasen-Simplex

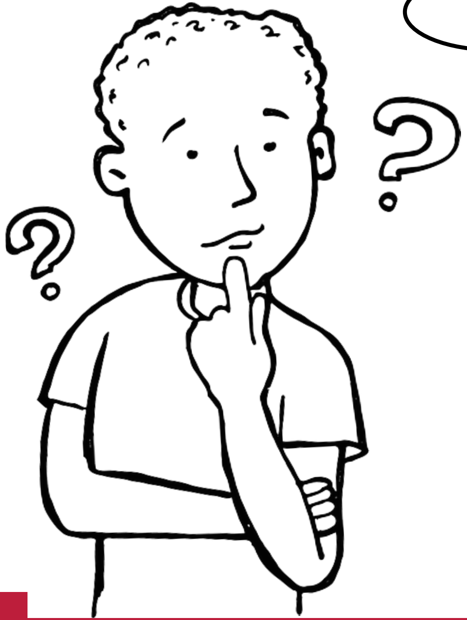


Basistausch



Fundamentalsatz und Pivotregeln

Überlegungen



Wir haben einen Algorithmus, der korrekt läuft… wenn er terminiert.

Terminiert der Simplex-Algorithmus immer?!

Falls nicht, was können wir tun?

Degenerierte Dictionaries und Pivots

$$
\begin{aligned}
\zeta &= 0 + x_1 + x_2 \\
w_1 &= 5 - 2x_1 + 3x_2 \\
w_2 &= 7 - 3x_1 - 7x_2 \\
w_3 &= 0 \qquad\quad - x_2
\end{aligned}
$$

$$
\begin{aligned}
\zeta &= 0 + x_1 - w_3 \\
w_1 &= 5 - 2x_1 + 3w_3 \\
w_2 &= 7 - 3x_1 - 7w_3 \\
x_2 &= 0 \qquad\quad - w_3
\end{aligned}
$$

Ein solches Dictionary heißt **degeneriert**. Es enthält ein $\bar{b}_i = 0$.

Dadurch entstehen **degenerierte Pivots**: Für ein $k \in \mathcal{B}$ existiert ein $i \in \mathcal{N}$ mit

$$\frac{\bar{b}_i}{\bar{a}_{ik}} = 0$$

Frage: Welche Pivots sind hier degeneriert und welche nicht?

x_1 nicht degeneriert, x_2 ist degeneriert.

Degenerierte Dictionaries und Pivots II

$$\begin{aligned} \zeta &= 0 + x_1 + x_2 \\ w_1 &= 5 - 2x_1 + 3x_2 \\ w_2 &= 7 - 3x_1 - 7x_2 \\ w_3 &= 0 \phantom{{}+ 3x_1} - x_2 \end{aligned}$$

$$\begin{aligned} \zeta &= 0 + x_1 - w_3 \\ w_1 &= 5 - 2x_1 + 3w_3 \\ w_2 &= 7 - 3x_1 - 7w_3 \\ x_2 &= 0 \phantom{{}+ 3x_1} - w_3 \end{aligned}$$

Durch den Pivotschritt hat sich der Zielfunktionswert nicht verändert!

Die Repräsentation des Wertes ist anders.
Die Lösung bleibt aber identisch:
$(x_1, x_2, w_1, w_2, w_3) = (0, 0, 5, 7, 0)$

- Macht das irgendwelche Probleme?
- Wie lange kann man im schlimmsten Fall den gleichen Wert behalten?

Degenerierte Dictionaries und Pivots II

$$\begin{array}{rrrr} \zeta = & 0\ + & x_1\ - & w_3 \\ w_1 = & 5\ - & 2x_1\ + & 3w_3 \\ w_2 = & 7\ - & 3x_1\ - & 7w_3 \\ x_2 = & 0 & - & w_3 \end{array}$$

$$\begin{array}{rrrr} \zeta = & \frac{7}{3}\ - & \frac{1}{3}w_2\ - & \frac{7}{3}w_3 \\ w_1 = & \frac{1}{3}\ - & \frac{2}{3}w_2\ - & \frac{10}{3}w_3 \\ x_1 = & \frac{7}{3}\ - & \frac{2}{3}w_2\ - & \frac{7}{3}w_3 \\ x_2 = & 0 & - & w_3 \end{array}$$

Ein weiterer Pivotschritt hat zwar die Degeneration nicht aufgelöst, ist aber bei einer optimalen Lösung angekommen!

Das passiert im Normalfall, dass wir trotz Degeneration in einem Optimum landen. Aber!

Es kann vorkommen, dass über degenerierte Pivots wieder Dictionaries erreicht werden, die bereits vorkamen und nie eine optimale Lösung gefunden wird.

Dieses Verhalten wird **zykeln** (cycling) genannt.

Zykeln

Um zykeln zu vermeiden, werden spezielle Regeln eingeführt. Aber sogar die folgende Regel kann zum Zykeln führen:

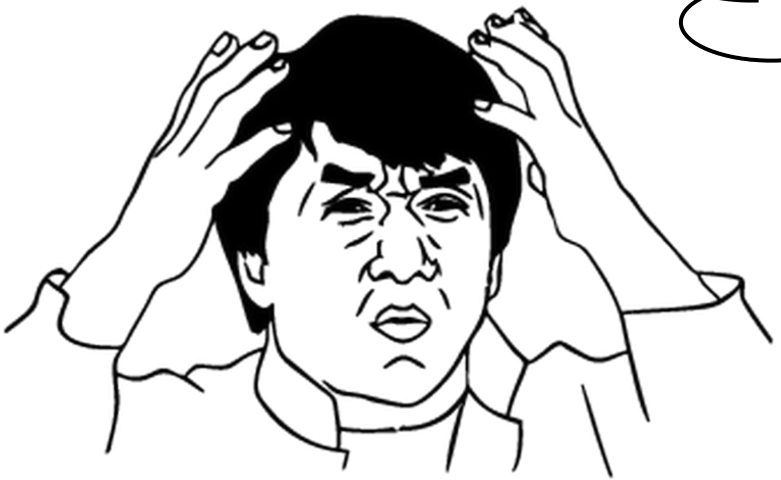
- Wähle als Nichtbasisvariable die mit dem größten positiven Koeffizienten $\bar{c}_k$.
- Wähle die Basisvariable nach lexikographischer Ordnung, wenn mehrere zur Auswahl stehen. (Dabei in der Regel Schlupfvariablen als letztes, also $x_1, \dots, x_n, w_1, \dots, w_m$)

$$
\begin{array}{rcrrrrr}
\zeta & = & & + x_1 & - 2x_2 & & - 2x_4 \\
\hline
w_1 & = & & - 0.5x_1 & + 3.5x_2 & + 2x_3 & - 4x_4 \\
w_2 & = & & - 0.5x_1 & + x_2 & + 0.5x_3 & - 0.5x_4 \\
w_3 & = & 1 & - x_1 & & &
\end{array}
$$

Probiert es selbst aus; nach 6 Iterationen sollte man wieder bei diesem Dictionary ankommen (ggf. sind die Spalten / Zeilen permutiert).

Und nun?

Was machen wir jetzt?!



Pertubation

Wenn ein $\bar{b}_i = 0$ ist, dann pertubiere es und setze $\bar{b}_i \coloneqq \bar{b}_i + \epsilon_i$.

Wenn viele davon existieren, pertubiere alle, sodass gilt:
$0 < \epsilon_k \ll \epsilon_{k-1} \ll \cdots \ll \epsilon_1 \ll$ andere Nicht-Null-Werte

Dabei im Hinterkopf zu behalten:
- Keine Linearkombination der ϵ_i kann während des Simplex-Algorithmus Größenordnungen wie die Daten im Problem annehmen.
- Niedrigere ϵ_j können nicht zu höheren ϵ_i, d.h. $i < j$, eskalieren.

Schauen wir uns ein Beispiel an!

Pertubationsbeispiel

$$\begin{array}{rcrrr}
\zeta & = & 0 + & 6x_1 + & 4x_2 \\
w_1 & = & 0 + & 9x_1 + & 4x_2 \\
w_2 & = & 0 - & 4x_1 - & 2x_2 \\
w_3 & = & 1 & - & x_2
\end{array}$$

Notiz: Man kann auch einfach jede Zeile pertubieren.

$$\begin{array}{rclllrr}
\zeta & = & 0 + & & & 6x_1 + & 4x_2 \\
w_1 & = & 0 + & \epsilon_1 + & & 9x_1 + & 4x_2 \\
w_2 & = & 0 + & & \epsilon_2 - & 4x_1 - & 2x_2 \\
w_3 & = & 1 & & & - & x_2
\end{array}$$

Iteration 1

$$\boxed{\begin{array}{rrrrrr} \zeta = & 0 + & & & 6x_1 + & 4x_2 \\ w_1 = & 0 + & \epsilon_1 + & & 9x_1 + & 4x_2 \\ w_2 = & 0 + & & \epsilon_2 - & 4x_1 - & 2x_2 \\ w_3 = & 1 & & & - & x_2 \end{array}}$$

Tausche x_1 mit w_2 aus.

$$\boxed{\begin{array}{rrrrrr} \zeta = & 0 + & & 1.5\epsilon_2 - & 1.5w_2 + & 1x_2 \\ w_1 = & 0 + & \epsilon_1 + & 2.25\epsilon_2 - & 2.25w_2 - & .5x_2 \\ x_1 = & 0 + & & .25\epsilon_2 - & .25w_2 - & .5x_2 \\ w_3 = & 1 & & & - & x_2 \end{array}}$$

Welches Pivot als nächstes?

Iteration 2

$$\boxed{\begin{array}{rcrrrrr} \zeta &=& 0\,+ & & 1.5\epsilon_2\,- & 1.5w_2\,+ & 1x_2 \\ w_1 &=& 0\,+ & \epsilon_1\,+ & 2.25\epsilon_2\,- & 2.25w_2\,- & .5x_2 \\ x_1 &=& 0\,+ & & .25\epsilon_2\,- & .25w_2\,- & .5x_2 \\ w_3 &=& 1 & & & - & x_2 \end{array}}$$

Tausche x_2 mit x_1 aus, da $\epsilon_2 \ll \epsilon_1$.

$$\boxed{\begin{array}{rcrrrrr} \zeta &=& 0\,+ & & 2\epsilon_2\,- & 2w_2\,- & 2x_1 \\ w_1 &=& 0\,+ & \epsilon_1\,+ & 2\epsilon_2\,- & 2w_2\,+ & x_1 \\ x_2 &=& 0\,+ & & .5\epsilon_2\,- & .5w_2\,- & 2x_1 \\ w_3 &=& 1\,- & & .5\epsilon_2\,+ & .5w_2\,+ & 2x_1 \end{array}}$$

Optimale Lösung

$$\boxed{\begin{aligned} \zeta &= 0 - 2w_2 - 2x_1 \\ w_1 &= 0 - 2w_2 + x_1 \\ x_2 &= 0 - .5w_2 - 2x_1 \\ w_3 &= 1 - .5w_2 + 2x_1 \end{aligned}}$$

Lösche die ϵ_i wieder, um das nicht-pertubierte Dictionary zu erhalten.

Wir haben die ϵ_i wie Symbole benutzt, wodurch wir eine präzise Beschreibung der Rangfolge der Variablen erhalten.

Die Methode wird auch **lexicographic method** genannt.

Theorem: Der Simplex-Algorithmus terminiert immer mit der lexicographic Method.

Beweisskizze

Zu zeigen: Es wird kein degeneriertes Dictionary erzeugt.

$$\begin{bmatrix} \epsilon_1 & 0 & \cdots & 0 \\ 0 & \epsilon_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \epsilon_m \end{bmatrix} \xrightarrow[\text{Simplex-Iterationen}]{\text{Bel. viele}} \begin{bmatrix} r_{11}\epsilon_1 & r_{12}\epsilon_2 & \cdots & r_{1m}\epsilon_m \\ r_{21}\epsilon_1 & r_{22}\epsilon_2 & \cdots & r_{2m}\epsilon_m \\ \vdots & \vdots & \ddots & \vdots \\ r_{m1}\epsilon_1 & r_{m2}\epsilon_2 & \cdots & r_{mm}\epsilon_m \end{bmatrix}$$

Nur Hauptdiagonale hat Einträge
➔ Rang m

Diese Matrix muss immer noch Rang m besitzen, da wir nur reversible Pivotoperationen durchgeführt haben.

Daraus folgt: Jede Zeile muss ein $r_{ij} \neq 0$ besitzen.

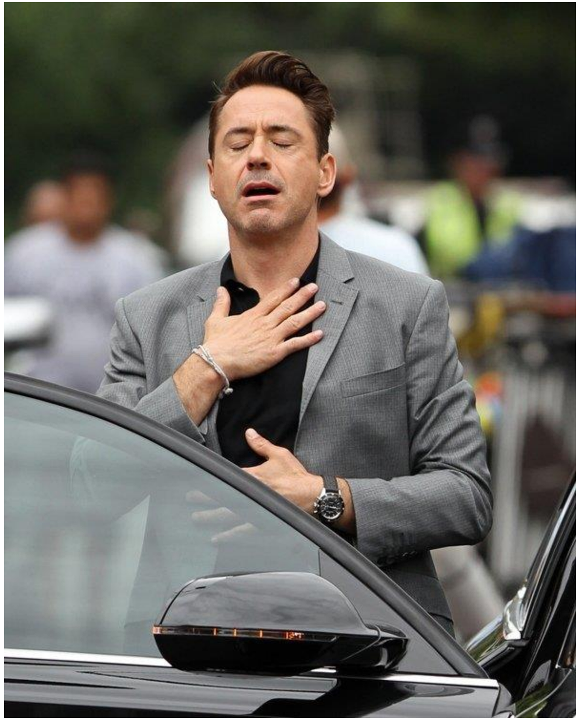
Daraus wiederum: Keine Zeile ist degeneriert (per Annahme, dass sich die ϵ_i nicht gegenseitig aufheben können).

Daher: Kein Dictionary wird degeneriert sein!

Phew!

Wir können also dafür sorgen, dass der Simplex-Algorithmus immer terminiert!

Damit können wir uns mögliche Ausgaben nach Terminierung anschauen!



Fundamentalsatz

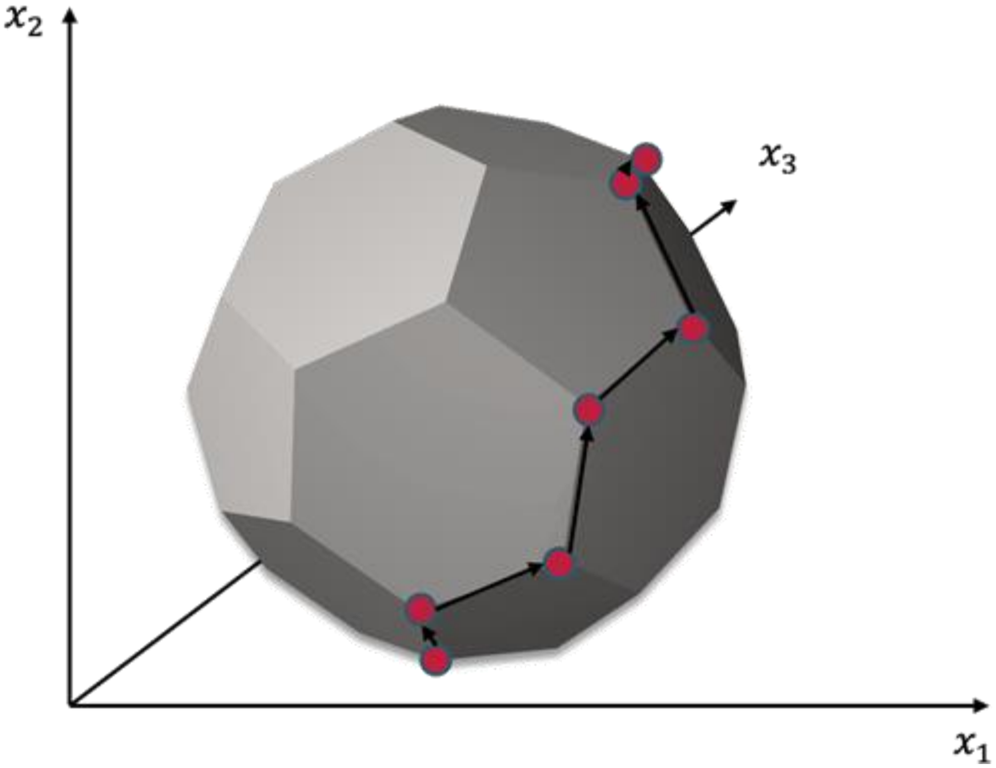
Theorem (Fundamentalsatz):
Für jedes LP in Standardform, gelten die folgenden Aussagen:

- Wenn es keine optimale Lösung gibt, ist das LP infeasible oder unbeschränkt.
- Wenn es eine Lösung gibt, dann existiert es eine Basislösung.
- Wenn es eine optimale Lösung gibt, dann existiert eine optimale Basislösung.

Basislösungen in der Geometrie

Die Extrempunkte des Polytops entsprechen Basislösungen!

Zum Nachdenken:
Wie sieht das mit degenerierten Basislösungen aus?



Andere Pivotregeln

Bland‘s Rule:
Wähle aus allen möglichen Pivots immer den mit dem kleinsten Index.

Simplex zykelt mit dieser Regel nicht! Benötigt aber ggf. länger zum Konvergieren.

Random Rule:
Wähle aus allen möglichen Pivots immer zufällig einen aus.

Greatest Increase Rule:
Pivotisiere so, dass die Zielfunktion am stärksten wächst.

Sehr rechenintensiv!

Okay… wir wissen, Simplex terminiert.
Aber wie lange dauert das?
In was messen wir die Komplexität?

Laufzeit Simplex-Algorithmus

Metriken:
- n, die Anzahl an Variablen
- m, die Anzahl an Constraints
- Anzahl an non-zero Koeffizienten

Rule-of-thumb: Je mehr 0en die Matrix enthält, desto einfacher wird es.

Worst-Case-Laufzeit hängt also eher von n und m ab.

Laufzeit Simplex-Algorithmus II

Die Laufzeit berechnet sich folgendermaßen:

$$T_{gesamt} = N_{iterationen} \cdot T_{iteration}$$

Also:

1. Wie viele Iterationen gibt es, und
2. Wie lange dauert eine Iteration?

Zu Frage 1: Wir gehen von Dictionary zu Dictionary ohne Wiederholung. Mit n+m (Schlupf-) Variablen von denen m in die Basis gehen, kann es maximal $\binom{n+m}{m}$ viele Dictionaries geben.

Im schlimmsten Fall, wenn n=m, sind das in etwa 2^{2n} Iterationen!


ABER WIR MÜSSEN NICHT
ALLE BASISLÖSUNGEN DURCH.


ABER WIR MÜSSEN NICHT
ALLE BASISLÖSUNGEN DURCH.
ODER?

Klee-Minty-Cube

Sei $1 = \beta_1 \ll \beta_2 \ll \cdots \ll \beta_n$. Betrachte folgendes LP

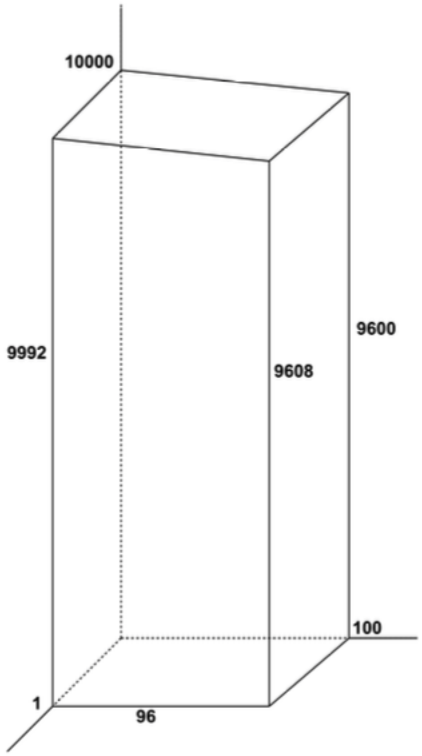
$$\begin{aligned}
\max_x \quad & \sum_{j=1}^{n} 2^{n-j} x_j - \frac{1}{2} \sum_{j=1}^{n} 2^{n-j} \beta_j \\
\text{s.t.} \quad & 2 \sum_{j=1}^{i-1} 2^{i-j} x_j + x_i \leq \sum_{j=1}^{i-1} 2^{i-j} \beta_j + \beta_i \quad i = 1, 2, \ldots, n \\
& x_j \geq 0 \qquad i = 1, 2, \ldots, n
\end{aligned}$$

Technische Universität Braunschweig

Beispiel

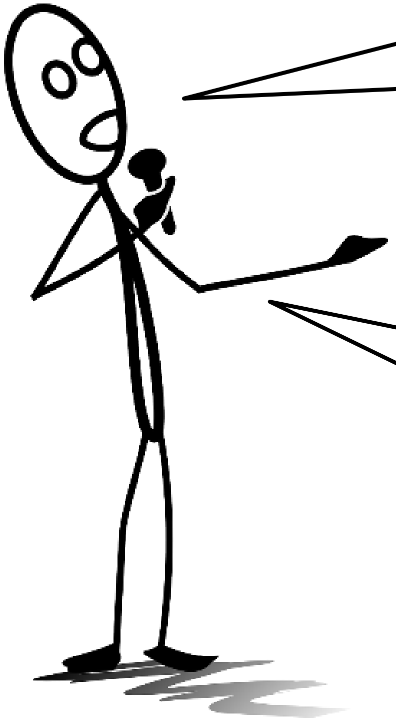
Mit $\beta_2 = 98, \beta_3 = 9800$ sieht der Lösungsraum so aus:

$$\begin{aligned} 1x_1 & & & \leq 1 \\ 4x_1 + & \quad 1x_2 & & \leq 100 \\ 8x_1 + & \quad 4x_2 + & \quad 1x_3 & \leq 10000 \\ x_1, & \quad x_2, & \quad x_3 & \geq 0 \end{aligned}$$



Lemma:
Mit der Größte-Pivot-Regel werden $2^n - 1$ Iterationen benötigt.

Simplex in der Praxis



Auch wenn Simplex **theoretisch** exponentielle Zeit besitzt, läuft der Algorithmus in der **Praxis** deutlich besser!

Außerdem hängt das Worst-Case-Beispiel sehr von der Pivot-Regel ab!

Nächste Woche