The Simplex Algorithm

Algorithm 1: Simplex Algorithm

Input: $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, and $c \in \mathbb{R}^n$

Output: $\tilde{x} \in \{x \in \mathbb{R}^n \mid Ax = b, x \geq 0\}$ maximizing $c^t x$ or the message that $\max\{c^t x \mid Ax = b, x \geq 0\}$ is unbounded or infeasible

1 Compute a feasible basis B;

2 If no such basis exists, stop with the message "INFEASIBLE";

3 Set $N = \{1, \ldots, n\} \setminus B$ and compute the feasible basic solution x for B;

4 Compute the simplex tableau $\begin{array}{rcrcl} x_B & = & p & + & Qx_N \\ z & = & z_0 & + & r^t x_N \end{array}$ for B;

5 **if** $r \leq 0$ **then**
  return $\tilde{x} = x$;

6 Choose $\alpha \in N$ with $r_\alpha > 0$;

7 **if** $q_{i\alpha} \geq 0$ **for all** $i \in B$ **then**
  return "UNBOUNDED";

8 Choose $\beta \in B$ with $q_{\beta\alpha} < 0$ and $\frac{p_\beta}{q_{\beta\alpha}} = \max\{\frac{p_i}{q_{i\alpha}} \mid q_{i\alpha} < 0, i \in B\}$;

9 Set $B = (B \setminus \{\beta\}) \cup \{\alpha\}$;

10 GOTO line 3;

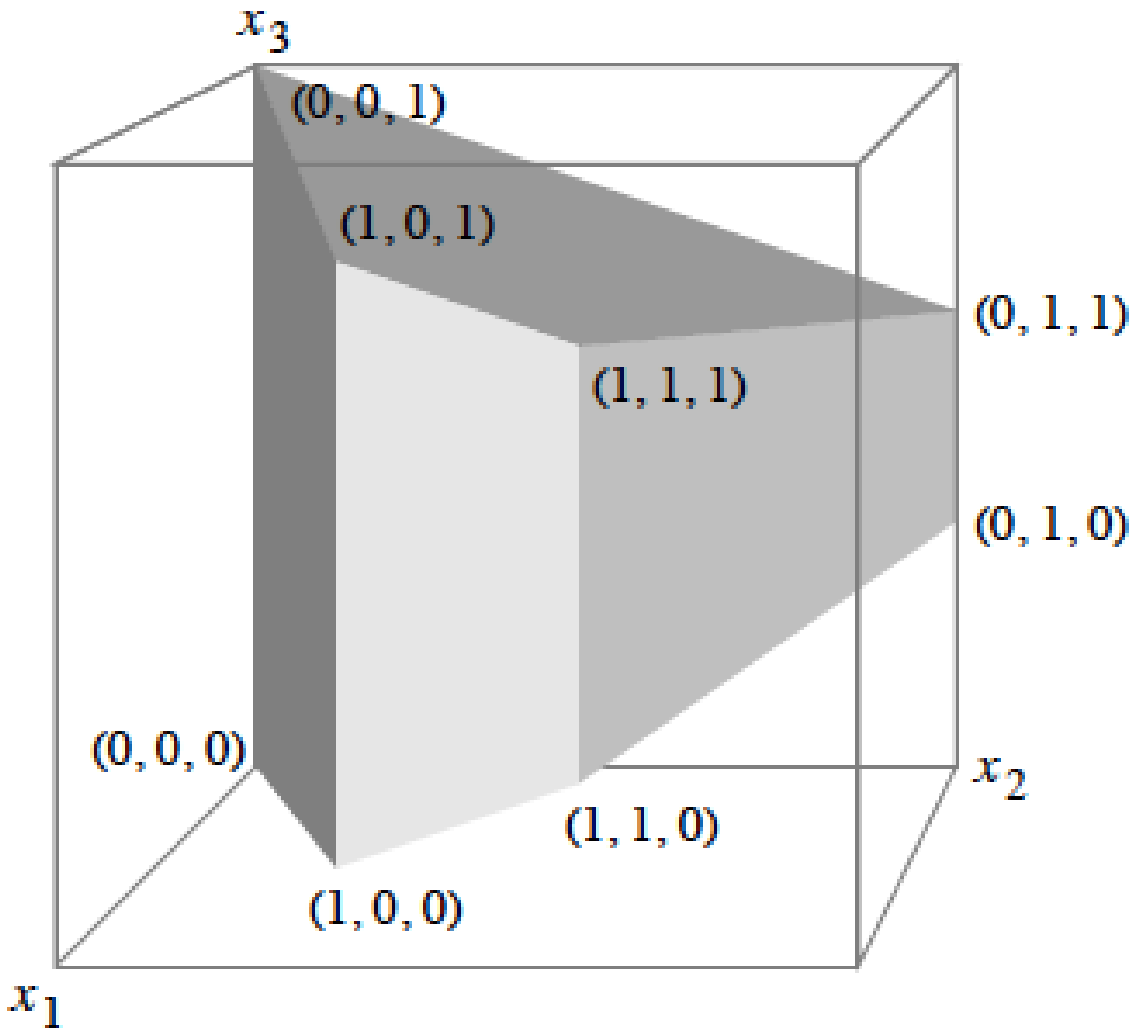
Klee-Minty-Würfel

$$
\begin{aligned}
\max\ & x_n & & \\
-x_1 &\leq 0 & & \\
x_1 &\leq 1 & & \\
\epsilon x_{j-1} - x_j &\leq 0 & & \text{for } j \in \{2, \ldots, n\} \\
\epsilon x_{j-1} + x_j &\leq 1 & & \text{for } j \in \{2, \ldots, n\}
\end{aligned}
$$

Der Lösungsraum dieses LPs heißt **Klee-Minty-Würfel**.

Mit der Bland-Regel betrachtet der SIMPLEX-ALGORITHMUS (beginnend mit der Ecke $(0, \ldots, 0)$) 2^n Basen.

Klee-Minty-Würfel



Effizienz des SIMPLEX-ALGORITHMUS

Schlechte Nachricht:

Für jede bisher angegebene Pivotregel konnten Beispiele gefunden werden, auf denen der SIMPLEX-ALGORITHMUS mit dieser Regel keine polynomielle Laufzeit hat.

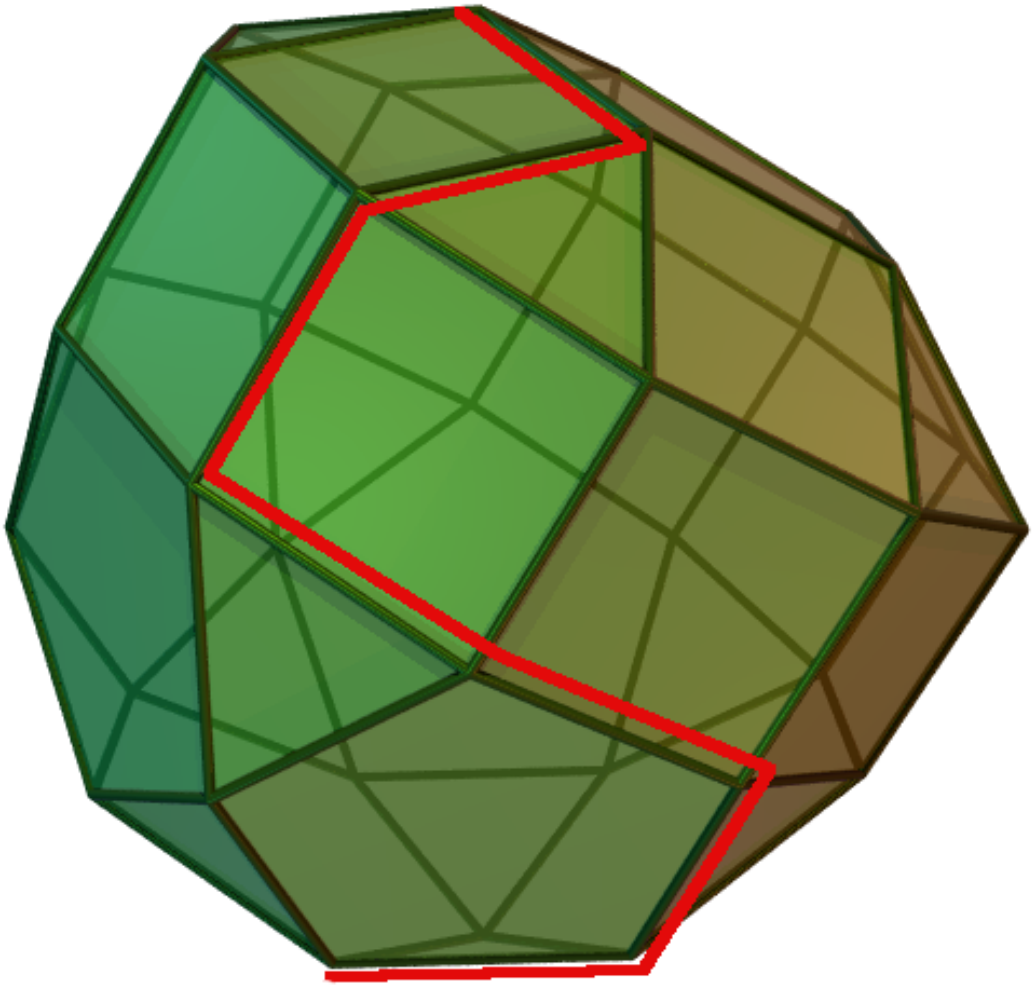
Effizienz des Simplex-Algorithmus

Kann man vielleicht allgemein zeigen, dass der Algorithmus nicht polynomiell ist?

Definition

Der **kombinatorische Durchmesser** eines spitzen Polyeders P ist der Durchmesser (d.h. der größte Abstand zwischen zwei Knoten) des ungerichteten Graphen G_P, wobei $V(G_P)$ die Menge der Ecken von P sei und zwei Knoten $v, w \in V(G_P)$ genau dann durch eine Kante in G_P verbunden sind, wenn es eine Fläche der Dimension 1 gibt, die v und w enthält.

Wege in G_p



Quelle: `https://de.wikipedia.org/`

Effizienz des Simplex-Algorithmus

Ohne Annahmen über die Startlösung ist der kombinatorische Durchmesser eine untere Schranke für die Laufzeit des SIMPLEX-ALGORITHMUS.

Aber: Es ist unbekannt, ob der der Durchmesser polynomiell (oder sogar linear) beschränkt in der Eingabegröße ist.

Revidierter Simplex-Algorithmus

$$\begin{array}{rcrcl} x_B & = & p & + & Qx_N \\ \hline z & = & z_0 & + & r^t x_N \end{array}$$

- Die zeitaufwendige explizite Berechnung des Simplex-Tableaus kann vermieden werden.
- Die $m \times (n-m)$-Matrix Q muss nicht ganz gespeichert werden. Es reicht, die Spalte von Q für den gewählten Index $\alpha \in N$ mit $r_\alpha > 0$ zu berechnen (Spaltenerzeugung/ column generation).
- A_B^{-1} muss nicht explizit bestimmt werden. Es reicht, Gleichungssysteme der Form $A_B y = d$ zu lösen. Dazu kann man z.B. eine LU-Zerlegung von A_B verwalten und geeignet aktualisieren.
- Der Simplex-Algorithmus mit diesen Anpassungen heißt Revidierter Simplex-Algorithmus.

Dualer Simplex-Algorithmus

Simplex-Tableau $T(B)$: $\begin{array}{rcrcl} x_B & = & p & + & Qx_N \\ \hline z & = & z_0 & + & r^t x_N \end{array}$

- Invariante während des (primalen) Simplex-Algorithmus: $p \geq 0$ (äquivalent zur (primalen) Zulässigkeit)
- Sobald $r \leq 0$ gilt, ist die Lösung optimal. Dann ist $\tilde{y} = A_B^{-t} c_B$ eine optimale Lösung des dualen LPs $\min\{b^t y \mid A^t y \geq c\}$.
- $r \leq 0$ ist äquivalent zur dualen Zulässigkeit.
- Im dualen Simplex-Algorithmus bleibt duale Zulässigkeit (also $r \leq 0$) stets garantiert, während primale Zulässigkeit (also $p \geq 0$) erst am Ende erreicht wird.
- Der duale Algorithmus hat dieselbe Effizienz wie der primale Algorithmus.
- Details: Siehe Übung.

Möglicher Vorteil des dualen Simplex-Algorithmus: Wenn neue Nebenbedingungen eingefügt werden, bleibt die duale Lösung zulässig und wir können mit ihr weiterrechnen.

Netzwerk-Simplex-Algorithmus

Netzwerk-Simplex-Algorithmus

- Idee: Wende den Simplex-Algorithmus an, um Min-Cost-Flow-Probleme zu lösen.
- Laufzeit wie im allgemeinen Fall: Keine polynomielle Laufzeit beweisbar, aber gut in der Praxis.
- Die Spezifikation der neuen Baumkanten und der Baumkanten, die entfernt werden, ist eine Pivotregel, die Kreiseln verhindert
  $\leadsto$ Siehe EDM-Vorlesung.

Polynomielle Algorithmen

Größen von Lösungen

Definiton von Darstellungsgrößen

- $n \in \mathbb{Z}$: $\text{size}(n) := 1 + \lceil \log(|n| + 1) \rceil$,
- $r = \frac{p}{q}$ mit $p, q \in \mathbb{Z}$ teilerfremd: $\text{size}(r) := \text{size}(p) + \text{size}(q)$,
- $x = (x_1, \ldots, x_n) \in \mathbb{Q}^n$: $\text{size}(x) := n + \sum_{i=1}^{n} \text{size}(x_i)$,
- $A = (a_{ij})_{\substack{i=1,\ldots,m \\ j=1,\ldots,n}} \in \mathbb{Q}^{m \times n}$: $\text{size}(A) := nm + \sum_{i=1}^{m} \sum_{j=1}^{n} \text{size}(a_{ij})$.

Annahme: Wir nehmen an, dass Brüche, die irgendwann auftreten, stets sofort mit dem EUKLIDISCHEN ALGORITHMUS gekürzt werden.

Größen von Lösungen

Satz:

Für $r_1, \ldots, r_n \in \mathbb{Q}$ gilt

(a) $\text{size}\left(\prod_{i=1}^{n} r_i\right) \leq \sum_{i=1}^{n} \text{size}(r_i)$

(b) $\text{size}\left(\sum_{i=1}^{n} r_i\right) \leq 2 \sum_{i=1}^{n} \text{size}(r_i)$

Größen von Lösungen

Satz:

Für $x, y \in \mathbb{Q}^n$ gilt

(a) $\text{size}(x + y) \leq 2(\text{size}(x) + \text{size}(y))$

(b) $\text{size}(x^t y) \leq 2(\text{size}(x) + \text{size}(y))$

Größen von Lösungen

Satz:

Für jede Matrix $A \in \mathbb{Q}^{n \times n}$ gilt $\text{size}(\det(A)) \leq 2\text{size}(A)$.

Größen von Lösungen

Satz:

Sei $\max\{c^t x \mid Ax \leq b\}$ ein zulässiges und beschränktes lineares Programm mit $A \in \mathbb{Q}^{m \times n}$ und $b \in \mathbb{Q}^m$. Dann gibt es eine optimale (rationale) Lösung x mit $\text{size}(x) \leq 4n(\text{size}(A) + \text{size}(b))$. Wenn $b = e_i$ oder $b = -e_i$ für einen Einheitsvektor e_i gilt, dann gibt es ein reguläre Teilmatrix A' von A und eine Optimallösung x mit $\text{size}(x) \leq 4n\text{size}(A')$.

Größen von Lösungen

Korollar:

Sei $\max\{c^t x \mid Ax \leq b\}$ ein zulässiges und beschränktes lineares Programm mit $A \in \mathbb{Q}^{m \times n}$ und $b \in \mathbb{Q}^m$. Dann gibt es eine optimale (rationale) Lösung x, sodass für jeden Nicht-Null-Eintrag x_j von x gilt: $|x_j| \geq 2^{-4n(\text{size}(A)+\text{size}(b))}$.

Gauss-Elimination

Wir wollen ein Gleichungssystem $Ax = b$ lösen.

Transformiere A dazu in eine obere rechte Dreicksmatrix mit folgenden erlaubten Schritten:

- Addiere ein Vielfaches einer Zeile zu einer anderen.
- Vertausche zwei Spalten.
- Vertausche zwei Zeilen.

Wir wissen aus Alma I: Es reichen dabei $O(mn(\text{rank}(A) + 1))$ elementare Operationen aus.

Ziel: Zeige polynomielle Laufzeit.