KAPITEL 8 — DAS SIMPLEXVERFAHREN

F. VALLENTIN, A. GUNDERT

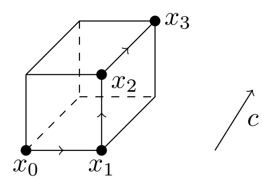
In diesem Kapitel lernen wir eine weitere Methode kennen, lineare Programme zu lösen: den Simplexalgorithmus.
Ziel: Löse (LP)

$$\text{(LP)} \qquad \begin{aligned} p^* = \max\ & c^T x \\ & x \in \mathbb{R}^n \\ & Ax \leq b. \end{aligned}$$

Wissen: Angenommen $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ ist ein Polytop. Dann wird das Maximum an einer Ecke von P angenommen ($\rightarrow$ Kapitel 5.5).

Geometrische Idee: Finde eine Sequenz von Ecken $x_0, x_1, \ldots, x_N \in P$, so dass

$$c^T x_0 \leq c^T x_1 \leq \ldots \leq c^T x_N = p^*.$$



1. Simplexalgorithmus mit bekannter Startecke

Zunächst treffen wir die spezielle Annahme, dass das Polyeder P eine Ecke x_0 besitzt, die wir kennen. Später beseitigen wir diese spezielle Annahme.
Simplexalgorithmus

- Wähle ein Teilsystem $A_0 x \leq b_0$ von $Ax \leq b$ mit einer regulären quadratischen Matrix A_0, wobei $A_0 x_0 = b_0$.
- Bestimme $u \in \mathbb{R}^m$ mit $c^T = u^T A$ und $u_i = 0$, falls Zeile i von A nicht zu A_0 gehört. Dazu berechne $c^T A_0^{-1}$ und füge Nullen an den entsprechenden Stellen hinzu.

1.Fall: $u \geq 0$.
Dann ist x_0 optimal, weil u eine optimale duale Lösung ist. Denn:

$$\begin{aligned} c^T x_0 = u^T A x_0 = u^T b &\geq \min\{y^T b : y \geq 0, y^T A = c^T\} \\ &= \max\{c^T x : Ax \leq b\}. \end{aligned}$$

Date: 14. Juli 2014.

2.Fall: $u \not\geq 0$.
Sei i der kleinste Index mit $u_i < 0$.

$$\text{Wähle } y \in \mathbb{R}^n \text{ mit } a^T y = 0 \text{ für alle Zeilen } a^T \text{ von } A_0 \text{ mit } a^T \neq a_i^T \text{ und } a_i^T y = -1.$$

(y ist die entsprechende Spalte von $-A_0^{-1}$).

2.a) $a^T y \leq 0$ für alle Zeilen a^T von A.
Dann ist $x_0 + \lambda y \in P \; \forall \lambda \geq 0$. Desweiteren ist

$$\begin{aligned} c^T(x_0 + \lambda y) &= c^T x_0 + \lambda c^T y \\ &= c^T x_0 + \lambda \underbrace{u^T A y}_{=-u_i} \\ &= c^T x_0 - \lambda u_i \to +\infty \text{ für } \lambda \to \infty. \end{aligned}$$

Das heißt das LP ist unbeschränkt.

2.b) $a^T y > 0$ für eine Zeile a^T von A.
Setze

$$\begin{aligned} \lambda_0 &= \max\{\lambda : x + \lambda y \in P\} \\ &= \min\{\frac{b_j - a_j^T x_0}{a_j^T y} : j = 1, \ldots, m, \; a_j^T y > 0\}, \end{aligned}$$

wobei die Gleichheit von Maximum und Minimum im Beweis des Theorems von Minkowski gezeigt wurde. Sei j der kleinste Index, an dem das Minimum angenommen wird. Definiere

$$\begin{aligned} A_1 &= \text{Matrix, die man aus } A_0 \text{ erhält,} \\ &\quad \text{indem man Zeile } a_i^T \text{ durch Zeile } a_j^T \text{ austauscht.} \\ x_1 &= x_0 + \lambda y. \end{aligned}$$

Dann gilt $A_1 x_1 = b_1$.

- Gehe zum Anfang mit A_1, x_1, anstelle von A_0, x_0.
- Wiederhole diese Schritte, bis $u \geq 0$ oder bis klar ist, dass das LP unbeschränkt ist.

Satz 1.1. *Der Simplexalgorithmus terminiert nach endlich vielen Schritten.*

Beweis. Wir bezeichnen die Variablen im k-ten Schritt mit $A_k, x_k, u_k, y_k, \lambda_{0,k}$. Es gilt

$$\begin{aligned} &c^T x_0 \leq c^T x_1 \leq \ldots, \text{ und} \\ &c^T x_k = c^T x_{k+1} \Leftrightarrow x_k = x_{k+1}, \end{aligned}$$

weil

$$c^T x_{k+1} = c^T(x_k + \lambda_{0,k} y_k) \quad \text{mit } \lambda_{0,k} \geq 0$$

und

$$c^T y_k = (-u_k)_i > 0.$$

Angenommen der Algorithmus landet in einer Endlosschleife. Dann gibt es k, l mit $k < l$ und $A_k = A_l$, weil es nur endlich viele verschiedene Teilmatrizen von A gibt. Dann

$$c^T x_k = c^T x_l, \text{ also } x_k = x_{k+1} = \ldots = x_l.$$

Sei r der größte Index, so dass die Zeile a_r^T in einer Iteration aus A_t genommen wird, wobei $t = k, k+1, \ldots, l$. Dies passiere in Schritt p. Weil $A_k = A_l$, gibt es ein q, so dass a_r^T wieder in A_q aufgenommen wird. Dann

$$k \leq p < q < l.$$

Dann gilt für $j > r$

$$a_j^T \text{ kommt in } A_p \text{ vor} \Longleftrightarrow a_j^T \text{ kommt in } A_q \text{ vor.}$$

Es gilt

$$u_p^T A y_q = c^T y_q > 0.$$

Also gibt es ein j mit $(u_p)_j(a_j^T y_q) > 0$. Aber:

1.Fall: a_j^T gehört nicht zu A_p. Dann $(u_p)_j = 0$, Widerspruch.
2.Fall: a_j^T gehört zu A_p.
- a) $j > r$: Dann $a_j^T y_q = 0$, Widerspruch.
- b) $j = r$: Dann $(u_p)_j < 0$ und $a_j^T y_q > 0$, Widerspruch.
- c) $j < r$: Dann $(u_p)_j \geq 0$ und $a_j^T y_q \leq 0$, Widerspruch.

□

2. Simplexalgorithmus ohne Startecke

Jetzt beschäftigen wir uns mit der Frage, wie man den Simplexalgorithmus startet, wenn man keine Ecke x_0 kennt.

OBdA: Das LP ist von der Form

$$\max\{c^T x : x \geq 0, Ax \leq b\}.$$

($\rightarrow$ Aufgabe 9.1: Dann besitzt $P = \{x \in \mathbb{R}^n : x \geq 0, Ax \leq b\}$ eine Ecke.)

Idee: Um eine Ecke von P zu finden, füge eine Extravariable hinzu und stelle ein neues LP auf, das eine offensichtliche Ecke besitzt und dessen optimale Lösung eine Ecke von P liefert.

Extravariable: $y \in \mathbb{R}^m,\ y \geq 0$.

Neues LP:

$$\min e^T y$$
$$\begin{bmatrix} A & -I_m \\ -I_n & 0 \\ 0 & -I_m \end{bmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \leq \begin{pmatrix} b \\ 0 \\ 0 \end{pmatrix},$$

mit $e = (1, \ldots, 1)^T$.

Offensichtliche Ecke:

$$x = 0, \quad y_j = \begin{cases} 0, & \text{falls } b_j \geq 0 \\ -b_j, & \text{falls } b_j < 0 \end{cases}, \; j = 1, \ldots, m.$$

Dann ist $\begin{pmatrix} x \\ y \end{pmatrix} \in \mathbb{R}^{n+m}$ eine Ecke von

$$P' = \{\begin{pmatrix} x \\ y \end{pmatrix} : Ax - y \leq b, x \geq 0, y \geq 0\},$$

weil $\begin{pmatrix} x \\ y \end{pmatrix} \in P'$ und rang $\begin{bmatrix} A & -I_m \\ -I_n & 0 \\ 0 & -I_m \end{bmatrix}_{\begin{pmatrix} x \\ y \end{pmatrix}} = n + m.$

Jetzt kann man den Simplexalgorithmus mit der Startecke $\begin{pmatrix} x \\ y \end{pmatrix}$ verwenden, um das neue LP zu lösen. Sei $\begin{pmatrix} x^* \\ y^* \end{pmatrix}$ die Ecke von P', die der Algorithmus liefert.

1.Fall: $e^T y^* > 0$.
Dann ist das Orginal-LP ungültig, denn

$$\nexists x \geq 0 : Ax \leq b, \text{ da } \exists j : y_j^* > 0 \text{ und } (Ax - y^*)_j \leq b_j.$$

2.Fall: $e^T y^* = 0$.
Dann $y^* = 0$ und x^* ist eine Ecke von P, weil $x^* \in P$ und rang $\begin{bmatrix} A \\ -I \end{bmatrix}_{x^*} = n.$

3. Zur praktischen und theoretischen Effizienz des Simplexalgorithmus

+ sehr schnell bei vielen praxisrelevanten Eingaben.
+ sehr gute Implementationen erhältlich (CPLEX, gurobi).
− Klee-Minty-Würfel (1972): Beispiel, dass der Algorithmus exponentiell viele Schritte im worst case benötigt.
+ Spielman-Teng (2004): „smoothed analysis“: Algorithmus ist polynomiell, falls die Eingabe leicht, zufällig „gestört“ist.
∘ offenes Problem („polynomielle Hirsch-Vermutung“): Ist der maximale Abstand zwischen zwei Ecken polynomiell in m, n?

Prof. Dr. F. Vallentin, Dr. A. Gundert, Mathematisches Institut, Universität zu Köln, Weyertal 86–90, 50931 Köln, Deutschland
E-mail address: frank.vallentin@uni-koeln.de, anna.gundert@uni-koeln.de