22) Lineare Programmierung

Allgemeine lineare Programme in Standard- und Schlupfform

Gegeben: lineare (affine) Zielfunktion

$$f(x_1, x_2, \ldots, x_n) = a_1 x_1 + \cdots + a_n x_n (+\nu), \qquad a_i \in \mathbb{R}.$$

Nebenbedingungen: lineare Ungleichungen

$$g_i(x_1, \ldots, x_n) \leq b_i, \qquad i = 1, \ldots, m.$$

Gesucht: $x_1, \ldots, x_n$, sodass $f(x_1, \ldots, x_n)$ max! (min!)

Bemerkung: Es geht auch $g_i(x_1, \ldots, x_n) \geq b_i$ oder $g_i(x_1, \ldots, x_n) = b_i$.

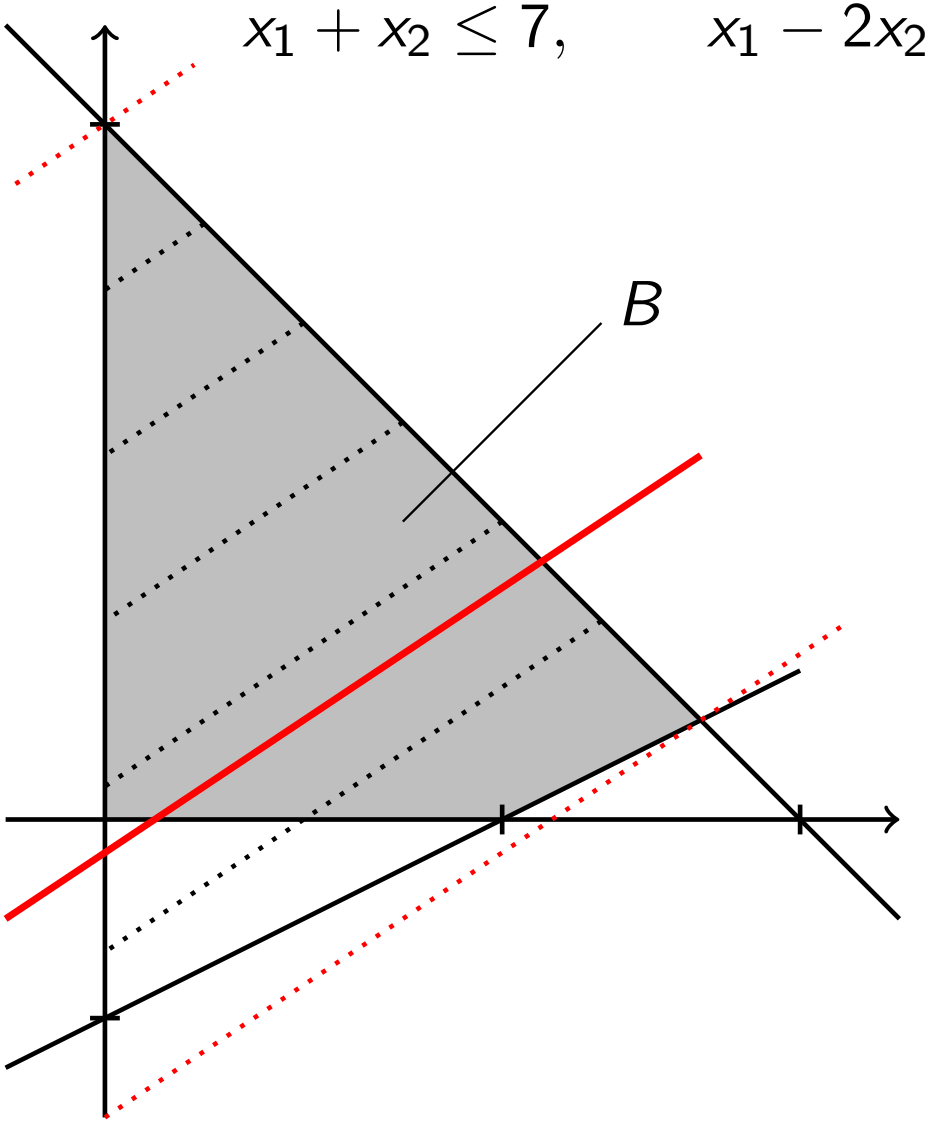
Definition

Jede Wahl $(x_1, \ldots, x_n)$, die alle Nebenbedingungen erfüllt, heißt zulässige Lösung des linearen Programms.

Beispiel:

$f(x_1, x_2) = 2x_1 - 3x_2$ max! mit Nebenbedingungen

$$x_1 + x_2 \leq 7, \qquad x_1 - 2x_2 \leq 4, \qquad x_1, x_2 \geq 0.$$



Graphische Lösung:

$$f(x_1, x_2) = 2x_1 - 3x_2 = z$$

ist Gerade mit Anstieg $\frac{2}{3}$;

$g \cap B$: Lösungsmenge zum Zielfunktionswert z.

B beschränkt, daher existiert eine Lösung.

Lineare Programmierung

Allgemein: n Variablen;
jede lineare Ungleichung beschreibt einen Halbraum.
Daher definieren die Nebenbedingungen ein **Simplex**.

Standardform eines linearen Programms:

$$\sum_{j=1}^{n} c_j x_j \quad \max! \qquad \text{NB: } \sum_{j=1}^{n} a_{i,j} x_j \leq b_i,\ 1 \leq i \leq m; \quad x_j \geq 0,\ 1 \leq j \leq n.$$

In Vektorform:

$$\begin{aligned} & \mathbf{c}^T \mathbf{x} \quad \max! \\ \text{NB:} \quad & A\mathbf{x} \leq \mathbf{b}, \quad \text{komponentenweise,} \\ & \mathbf{x} \geq \mathbf{0}, \quad \text{komponentenweise.} \end{aligned}$$

Simplex-Algorithmus

Eingabe: lineares Programm, Ausgabe: optimale Lösung.

- Beginne bei einer Ecke $\mathbf{x}_0$ des Simplex.
- Gehe zu benachbarter Ecke $\mathbf{x}_1$ (entlang einer Kante), sodass $f(\mathbf{x}_1) > f(\mathbf{x}_0)$.
- Terminiere, wenn $\mathbf{x}_i$ ein lokales Maximum ist.

Bemerkung: Ein lokales Maximum ist auch ein globales Maximum, da f linear und B konvex ist.

Definition

Zwei lineare Programme P und P' sind äquivalent, i.Z. $P \leftrightarrow P'$, wenn für jede Lösung $\mathbf{x}$ von P mit $f(\mathbf{x}) = z$ eine Lösung $\mathbf{x}'$ von P' mit $f'(\mathbf{x}') = z$ existiert und umgekehrt.

Schlupfform:

Übersetzen der NB:

$$\sum_{j=1}^{n} a_{i,j}x_j \leq b_i \longrightarrow \begin{cases} s_i = b_i - \sum_{j=1}^{n} a_{i,j}x_j \\ s_i \geq 0 \end{cases} \qquad \begin{array}{l} s_i \text{ Schlupfvariable} \\ x_{n+i} := s_i. \end{array}$$

Dies führt auf die Schlupfform:

$$\nu + \sum_{j=1}^{n} c_j x_j \quad \max!$$

$$x_{n+i} = b_i - \sum_{j=1}^{n} a_{i,j}x_j, \qquad 1 \leq i \leq m,$$

$$x_1, \ldots, x_n, x_{n+1}, \ldots, x_{n+m} \geq 0.$$

$x_1, \ldots, x_n$ heißen freie Variablen,
$x_{n+1}, \ldots, x_{n+m}$ Schlupfvariablen (oder Basisvariablen).

Lineare Programmierung

Konversion in Standardform: gegeben sei ein lineares Programm P.

- $\left.\begin{array}{c}\min \to \max \\ \geq \to \leq\end{array}\right\} \to \cdot(-1)$
- Nebenbedingung enthält Gleichungen $g(x_1, \ldots, x_n) = b$: Ersetzen durch die Ungleichungen $-g(x_1, \ldots, x_n) \leq -b$ und $g(x_1, \ldots, x_n) \leq b$.
- $x_j \geq 0$ fehlt: $x_j = x'_j - x''_j$, $x'_j \geq 0, x''_j \geq 0$ hinzufügen; führt auf lineares Programm P'.

  Sei $\mathbf{x}'$ zulässige Lösung von P'. Dann liefert $x_j := x'_j - x''_j$ eine zulässige Lösung $\mathbf{x}$ von P;

  Sei $\mathbf{x}$ zulässige Lösung von P. Dann setzen wir $x'_j := x_j$ und $x''_j := 0$ und erhalten eine zulässige Lösung von P', falls $x_j \geq 0$, andernfalls setzen wir $x'_j := 0$ und $x''_j := -x_j$.

  Somit sind P und P' äquivalent.

Probleme als lineare Programme

Kürzeste Wege:

- Gegeben: $G = (V, E)$ gerichtet, $w : E \to \mathbb{R}$, $s,\ t \in V$.
- Gesucht: kürzester Weg $s \to t$.
- Lineares Programm:

$$\begin{aligned} & d_t \quad \max! \\ & d_v \leq d_u + w(u,v), \qquad (u,v) \in E, \\ & d_s = 0. \end{aligned}$$

$|V|$ Variable, $|E| + 1$ Nebenbedingungen.

Bemerkung: Sei $\bar{d}_v := d(s, v)$ für $v \in V$.
Dann gilt: $\bar{d}_v = \min_{u:(u,v)\in E} \{\bar{d}_u + w(u,v)\}$ und $\bar{d}_s = 0$, d.h.

$$\bar{d}_v = \max\{x \in \mathbb{R} \mid x \leq \bar{d}_u + w(u,v) \text{ für alle } u \text{ mit } (u,v) \in E\}.$$

Lineare Programmierung

Maximaler Fluss:

- Gegeben: Flussnetzwerk $G = (V, E, w, s, t)$.
- Gesucht: Fluss ϕ mit $|\phi| = \sum_{v \in \Gamma^+(s)} \phi(s, v)$ max!.

Lineares Programm:

$$\sum_{v \in \Gamma^+(s)} \phi(s, v) \quad \text{max!}$$

$$\begin{aligned}
\phi(u, v) &\leq w(u, v), && (u, v) \in E; \\
\phi(u, v) &\geq 0, && (u, v) \in E; \\
\sum_{v \in \Gamma^-(u)} \phi(v, u) &= \sum_{v \in \Gamma^+(u)} \phi(u, v), && u \in V \setminus \{s, t\}.
\end{aligned}$$

$|E|$ Variable, $2|E| + |V| - 2$ Nebenbedingungen.

Lineare Programmierung

Billigster Fluss:

- Gegeben: Flussnetzwerk $G = (V, E, w, s, t)$, Kostenfunktion $k : E \to \mathbb{R}$
- Gesucht: Billigster Fluss ϕ mit $|\phi| = d$.

Lineares Programm:

$$\sum_{(u,v)\in E} k(u,v)\phi(u,v) \quad \text{min!}$$

$$\begin{aligned}
\phi(u,v) &\le w(u,v), && (u,v) \in E;\\
\phi(u,v) &\ge 0, && (u,v) \in E;\\
\sum_{v\in\Gamma^-(u)} \phi(v,u) &= \sum_{v\in\Gamma^+(u)} \phi(u,v), && u \in V \setminus \{s,t\};\\
\sum_{v\in\Gamma^+(s)} \phi(s,v) &= d.
\end{aligned}$$

Eine Nebenbedingung mehr als vorher.

Der Simplex-Algorithmus

(i) Jeder Iteration wird eine Basislösung zugeordnet (leicht aus der Schlupfform bestimmbar):
Freie Variablen: $x_1 = x_2 = \cdots = x_n = 0$,
Schlupfvariable aus den Gleichungen der Nebenbedingungen.

(ii) Jede Iteration konvertiert Schlupfform in eine äquivalente Schlupfform, sodass der Zielfunktionswert der neuen Schlupfform (an der Basislösung) größer ist als der Zielfunktionswert der Basislösung der alten Schlupfform.

- Wähle freie Variable x,
  sodass mit wachsendem x auch f wächst.
- x wächst bis $y = 0$ für eine Schlupfvariable y.
- Rollen von x und y vertauschen. (Basiswechsel)

Beispiel

$$
\begin{aligned}
3x_1 + x_2 + 2x_3 \quad & \text{max!} \\
x_1 + x_2 + 3x_3 &\leq 30 \\
2x_1 + 2x_2 + 5x_3 &\leq 24 \\
4x_1 + x_2 + 2x_3 &\leq 36 \\
x_1,\, x_2,\, x_3 &\geq 0
\end{aligned}
$$

in Standardform. Die Schlupfform ist dann

$$
\begin{aligned}
z = \quad & 3x_1 + x_2 + 2x_3 \\
x_4 &= 30 - x_1 - x_2 - 3x_3 \\
x_5 &= 24 - 2x_1 - 2x_2 - 5x_3 \\
x_6 &= 36 - 4x_1 - x_2 - 2x_3 \\
x_1, x_2, x_3, x_4, x_5, x_6 &\geq 0,
\end{aligned}
$$

Als Matrix:

z	x_4	x_5	x_6	konst.	x_1	x_2	x_3
1	0	0	0	0	3	1	2
0	1	0	0	30	-1	-1	-3
0	0	1	0	24	-2	-2	-5
0	0	0	1	36	-4	-1	-2

Start: Basislösung $\mathbf{x} = (0, 0, 0, 30, 24, 36)$ mit $f(\mathbf{x}) = 0$.
Bem.: Basislösung ist zulässige Lösung. Nicht immer so!
Im Allgemeinen ist daher Initialisierung nötig, die zulässige Ecke des Simplex liefert.

Wähle freie Variable mit positiven Koeffizienten in der Zielfunktion, z.B. x_1.
Aus den NB: $x_1 \leq 30$, $x_1 \leq 12$, $x_1 \leq 9$.
Also $x_1 = 9$, $x_6 = 0$; dann letzte Gleichung nach x_1 auflösen und alle x_1 durch das Ergebnis ausdrücken. $x_1 = 9 - \frac{1}{4}x_2 - \frac{1}{2}x_3 - \frac{1}{4}x_6$

z	x_1	x_4	x_5	konst.	x_2	x_3	x_6
1	0	0	0	27	$1/4$	$1/2$	$-3/4$
0	1	0	0	9	$-1/4$	$-1/2$	$-1/4$
0	0	1	0	21	$-3/4$	$-5/2$	$1/4$
0	0	0	1	6	$-3/2$	-4	$1/2$

Basislösung: $\mathbf{x}' = (9, 0, 0, 21, 6, 0)$, $z = f(\mathbf{x}') = 27$.

x_2 oder x_3 wählbar, z.B. x_2.
Aus den NB: $x_2 \leq 36$, $x_2 \leq 28$, $x_2 \leq 4$. $\longrightarrow x_2 = 4$, $x_5 = 0$.
$x_2 = 4 - \frac{8}{3}x_3 - \frac{2}{3}x_5 + \frac{1}{3}x_0$

z	x_1	x_2	x_4		x_3	x_5	x_6
1	0	0	0	28	$-1/6$	$-1/6$	$-2/3$
0	1	0	0	8	$1/6$	$1/6$	$-1/3$
0	0	1	0	4	$-8/3$	$-2/3$	$1/3$
0	0	0	1	18	$-1/2$	$1/2$	0

z optimal!

$\mathbf{x}'' = (8, 4, 0, 18, 0, 0)$, $z = 28$.

Basiswechsel:

F ... Menge der Indizes der freien Variablen;

S ... Menge der Indizes der Schlupfvariablen;

$A = (a_{i,j})_{i \in S, j \in F}$, $\mathbf{b} = (b_i)_{i \in S}$, $\mathbf{c} = (c_j)_{j \in F}$.

Eingabe: $(F, S, A, \mathbf{b}, \mathbf{c}, \nu, g, k)$

Ausgabe: $(\hat{F}, \hat{S}, \hat{A}, \hat{\mathbf{b}}, \hat{\mathbf{c}}, \hat{\nu})$ ... neue Schlupfform

$$z = \nu + \sum_{j \in F} c_j x_j$$

$$x_i = b_i - \sum_{j \in F} a_{i,j} x_j, \qquad i \in S$$

$$x_j \geq 0, \qquad j \in F \cup S$$

$g \in S$, $k \in F$, nach $x_g \leftrightarrow x_k$ dann $g \in \hat{F}$, $k \in \hat{S}$.

Algorithm BW$(F, S, A, \mathbf{b}, \mathbf{c}, \nu, g, k)$

Sei $\hat{A}$ eine neue $m \times n$-Matrix % Koeffizienten der Gleichung für x_k
$\hat{b}_k := b_g / a_{g,k}$
for $j \in F \setminus \{k\}$ **do**
    $\hat{a}_{k,j} := a_{g,j} / a_{g,k}$
end for
$\hat{a}_{k,g} := 1 / a_{g,k}$
for $i \in S \setminus \{g\}$ **do** % Koeffizienten der übrigen NB
    $\hat{b}_i := b_i - a_{i,k} \hat{b}_k$
    for $j \in F \setminus \{k\}$ **do**
        $\hat{a}_{i,j} := a_{i,j} - a_{i,k} \hat{a}_{k,j}$
    end for
    $\hat{a}_{i,g} := -a_{i,k} \hat{a}_{k,g}$
end for
$\hat{\nu} := \nu + c_k \hat{b}_k$ % neue Zielfunktion
for $j \in F \setminus \{k\}$ **do**
    $\hat{c}_j := c_j - c_k \hat{a}_{k,j}$
end for
$\hat{c}_g := -c_k \hat{a}_{k,g}$
$\hat{F} := F \setminus \{k\} \cup \{g\}$
$\hat{S} := S \setminus \{g\} \cup \{k\}$
return $(\hat{F}, \hat{S}, \hat{A}, \hat{\mathbf{b}}, \hat{\mathbf{c}}, \hat{\nu})$

Zeilen 2–6 (Koeffizienten der Gleichung für x_k): x_k berechnen aus

$$x_g = b_g - \sum_{j \in F} a_{g,j} x_j \qquad \Rightarrow x_k = \frac{1}{a_{g,k}} \left(b_g - \sum_{j \in F \setminus \{k\}} a_{g,j} x_j - x_g \right).$$

Zeilen 7–12 (Koeffizienten der übrigen NB): Berechnung der neuen Koeffizienten für x_j (x_k einsetzen) aus

$$x_i = b_i - \sum_{j \in F \setminus \{k\}} a_{i,j} x_j - a_{i,k} x_k,$$

$$x_k = \hat{b}_k - \sum_{j \in F \setminus \{k\}} \hat{a}_{k,j} x_j - \hat{a}_{k,g} x_g.$$

Zeilen 14–18 (neue Zielfunktion): x_k einsetzen in

$$z = \nu + \sum_{j \in F \setminus \{k\}} c_j x_j + c_k x_k.$$

Lemma

Sei $a_{g,k} \neq 0$. Betrachte Aufruf $\mathrm{BW}(F, S, A, \mathbf{b}, \mathbf{c}, \nu, g, k)$ mit Ergebnis $(\hat{F}, \hat{S}, \hat{A}, \hat{\mathbf{b}}, \hat{\mathbf{c}}, \hat{\nu})$, $\bar{\mathbf{x}}$ sei Basislösung nach dem Aufruf. Dann gilt:

1. $\bar{x}_j = 0 \quad \forall j \in \hat{F}$
2. $\bar{x}_k = \frac{b_g}{a_{g,k}}$
3. $\bar{x}_i = b_i - a_{i,k}\hat{b}_k \quad \forall i \in \hat{S} \setminus \{k\}$

Beweis: Die Basislösung setzt alle freien Variablen auf 0. Daraus folgt 1.).

Wenn wir alle freien Variablen auf 0 setzen, gilt wegen $x_i = \hat{b}_i - \sum_{j \in \hat{F}} \hat{a}_{i,j} x_j$ für all $i \in \hat{S}$ dann $\bar{x}_i = \hat{b}_i$. Aus $k \in \hat{S}$ folgt $\bar{x}_k = \hat{b}_k = b_g / a_{g,k}$ (Zeile 2) und damit 2.).

Aus $i \in \hat{S} \setminus \{k\}$ folgt wegen $\hat{b}_i := b_i - a_{i,k}\hat{b}_k$ (Zeile 8) schließlich $\bar{x}_i = \hat{b}_i = b_i - a_{i,k}\hat{b}_k$, also 3.). □

Lineare Programmierung

Fragen zum Simplex-Algorithmus:

1. Wie stellen wir die Lösbarkeit fest?
2. Programm ist lösbar, aber die Basislösung nicht zulässig.
3. Beschränkt/unbeschränkt?
4. Auswahl von x_g und x_k?

Wir nehmen zunächst an, dass wir eine Prozedur INIT-SIMPLEX$(A, \mathbf{b}, \mathbf{c})$ haben, die zu einer Standardform

- eines unlösbaren linearen Programms mit der Ausgabe „unlösbar“ terminiert,
- und andernfalls eine Schlupfform mit zulässiger Basislösung ausgibt.

Algorithm SIMPLEX($A, \mathbf{b}, \mathbf{c}$)

1: $(F, S, A, \mathbf{b}, \mathbf{c}, \nu)$:=INIT-SIMPLEX($A, \mathbf{b}, \mathbf{c}$)
2: Sei Δ ein neuer Vektor mit m Einträgen
3: **while** $\exists j \in F : c_j > 0$ **do**
4: Wähle $k \in F$ mit $c_k > 0$
5: **for** $i \in S$ **do**
6: **if** $a_{i,k} > 0$ **then** $\Delta_i := b_i / a_{i,k}$ **else** $\Delta_i := \infty$
7: **end if**
8: **end for**
9: Wähle $g \in S$, sodass Δ_g minimal
10: **if** $\Delta_g = \infty$ **then**
11: **return** „unbeschränkt“
12: **else**
13: $(F, S, A, \mathbf{b}, \mathbf{c}, \nu)$:=BW($F, S, A, \mathbf{b}, \mathbf{c}, \nu, g, k$)
14: **end if**
15: **end while**
16: **for** $i = 1$ to n **do**
17: **if** $i \in S$ **then** $\bar{x}_i := b_i$ **else** $\bar{x}_i := 0$
18: **end if**
19: **end for**
20: **return** $(\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_n)$

Lineare Programmierung

Korrektheit: Folgendes ist zu zeigen:

- Falls SIMPLEX terminiert, dann erfolgt die
  1. Ausgabe einer neuen Schlupfform mit zulässiger Basislösung;
  2. oder das lineare Programm ist unbeschränkt.
- Der Algorithmus terminiert.
- Die ausgegebene Lösung ist optimal.
- INIT-SIMPLEX

Lemma

Sei $(A, \mathbf{b}, \mathbf{c})$ *ein lineares Programm in Standardform und* $\text{INIT-SIMPLEX}(A, \mathbf{b}, \mathbf{c})$ *gebe eine Schlupfform mit zulässiger Basislösung aus.*
Falls SIMPLEX *terminiert, dann gilt eine der beiden Aussagen:*

1. *Falls* SIMPLEX *eine Lösung ausgibt, dann ist es eine zulässige Lösung.*
2. *Das lineare Programm ist unbeschränkt.*

Beweis: Schleifeninvariante für die **while**-Schleife:

„Am Beginn jeder Iteration gilt:

1. Schlupfform ist äquivalent zur Schlupfform von INIT-SIMPLEX.
2. $\forall i \in S : b_i \geq 0$.
3. Basislösung ist zulässig.“

Initialisierung: (1) und (3): ✓
(2): Da INIT-SIMPLEX eine Schlupfform mit zulässiger Basislösung liefert, gilt für $i \in S$ $x_i \geq 0$ und für die Basislösung: $x_i = b_i$, also $b_i \geq 0$.

Aufrechterhaltung: Schleife bewirkt $x_g \leftrightarrow x_k$ durch Aufruf von BW. BW gibt äquivalente Schlupfform zurück, woraus (1) folgt.

(2): Annahme, am Beginn der **while**-Schleife gelte $b_i \geq 0$ für alle $i \in S$.
Z.z.: Am Ende (nach Ausführen von BW) gilt das immer noch.

Vor dem Basiswechsel haben wir $b_i, a_{i,j}, S$, danach $\hat{b}_i, \hat{a}_{i,j}, \hat{S}$.
$\hat{b}_k \geq 0$, denn $b_g \geq 0$ und $a_{g,k} > 0$:
Zeile 6: „**if** $a_{i,k} > 0 \ldots$“garantiert, dass $\Delta_i < \infty$ nur für $a_{i,k} > 0$,
Zeile 9: „Wähle $g \in S$ mit Δ_g minimal“
Zeile 2 von BW: $\hat{b}_k := b_g / a_{g,k}$.

Sei $i \in S \setminus \{g\}$. Dann gilt nach Zeile 8 in BW:

$$\hat{b}_i = b_i - a_{i,k}\hat{b}_k = b_i - a_{i,k}\frac{b_g}{a_{g,k}}.$$

1. Fall: $a_{i,k} > 0$. g so gewählt, dass $\frac{b_g}{a_{g,k}} \leq \frac{b_i}{a_{i,k}}$ für alle $i \in S$.
   Daraus folgt $\hat{b}_i \geq 0$.
2. Fall: $a_{i,k} \leq 0$. Das impliziert $\hat{b}_i \geq 0$, da $a_{g,k}, b_g > 0$.

(3): Zu zeigen: $x_i \geq 0$ für alle i.

- $i \in F$: ✓ (werden auf 0 gesetzt)
- $i \in S$: $x_i = \hat{b}_i - \sum_{j \in F} \hat{a}_{i,j} x_j = \hat{b}_i \geq 0$ wegen (2).

Terminierung: Zwei Möglichkeiten:

- **while**-Bedingung verletzt: Basislösung zulässig, wird ausgegeben.
- **return** „unbeschränkt“. Das bedeutet, dass $\Delta_i = \infty$ und daher $a_{i,k} \leq 0$ für alle $i \in S$.

Betrachte Lösung: $\bar{\mathbf{x}}$ mit

$$\bar{x}_i = \begin{cases} \infty & \text{für } i = k, \\ 0 & \text{für } i \in F \setminus \{k\}, \\ b_i - \sum_{j \in F} a_{i,j} \bar{x}_j & \text{für } i \in S \end{cases}$$

Daraus folgt $\bar{x}_i \geq 0$ für alle $i \in F$.

Für $i \in S$ haben wir

$$\bar{x}_i = b_i - \sum_{j \in F} a_{i,j}\bar{x}_j = b_i - a_{i,k}\bar{x}_k \geq 0,$$

da $b_i \geq 0$ und $a_{i,k} \leq 0$.

Wegen $c_k > 0$ und $z = \nu + \sum_{j \in F} c_j \bar{x}_j = \nu + c_k \bar{x}_k = \infty$ ist das lineare Programm unbeschränkt. □

Ad **while**-Bedingung verletzt: $z = \nu + \sum_{j \in F} c_j x_j$ und alle $c_j < 0$. Daher ist z ein Maximum.

Terminierung des SIMPLEX-Algorithmus:

z-Wert ist monoton wachsend, möglicherweise aber nicht strikt.

Beispiel:

$$z = x_1 + x_2 + x_3$$
$$x_4 = 8 - x_1 - x_2$$
$$x_5 = x_2 - x_3$$

Wähle $x_k = x_1 \implies x_g = x_4$

$$z = 8 + x_3 - x_4$$
$$x_1 = 8 - x_2 - x_4$$
$$x_5 = x_2 - x_3$$

Wir müssen $x_k = x_3$ wählen. $\implies x_g = x_5$

$$z = 8 + x_2 - x_4 - x_5$$
$$x_1 = 8 - x_2 - x_4$$
$$x_3 = x_2 - x_5$$

$z = 8$ bleibt unverändert!

Terminierung des SIMPLEX-Algorithmus:

Beispiel 2:

$$z = 2,3x_1 + 2,15x_2 - 13,55x_3 - 0,4x_4$$
$$x_5 = -0,4x_1 - 0,2x_2 + 1,4x_3 + 0,2x_4$$
$$x_6 = 7,8x_1 + 1,4x_2 - 7,8x_3 - 0,4x_4$$

Wähle $x_k = x_1,\ x_g = x_5$

$$z = x_2 - 5,5x_3 + 0,75x_4 - 5,75x_5$$
$$x_1 = -0,5x_2 + 3,5x_3 + 0,5x_4 - 2,5x_5$$
$$x_6 = -2,5x_2 + 19,5x_3 + 3,5x_4 - 19,5x_5$$

Wähle $x_k = x_2,\ x_g = x_6$

$$z = 2,3x_3 + 2,15x_4 - 13,55x_5 - 0,4x_6$$
$$x_1 = -0,4x_3 - 0,2x_4 + 1,4x_5 + 0,2x_6$$
$$x_2 = 7,8x_3 + 1,4x_4 - 7,8x_5 - 0,4x_6$$

SIMPLEX kreist nach weiteren 4 Iterationen!

Lemma

Sei $(A, \mathbf{b}, \mathbf{c})$ eine Standardform und S vorgegeben. Dann ist die Schlupfform eindeutig bestimmt.

Beweis: Annahme, es existieren 2 verschiedene Schlupfformen. Wir haben $F = \{1, 2, \ldots, m+n\} \setminus S$.

$$z = \nu + \sum_{j \in F} c_j x_j \qquad z = \nu' + \sum_{j \in F} c'_j x_j$$

$$x_i = b_i - \sum_{j \in F} a_{i,j} x_j \qquad x_i = b'_i - \sum_{j \in F} a'_{i,j} x_j$$

Daraus folgt

$$0 = (b_i - b'_i) - \sum_{j \in F} (a_{i,j} - a'_{i,j}) x_j, \quad i \in S,$$

für alle x_j. Daher gilt $b_i = b'_i$ und $a_{i,j} = a'_{i,j}$.

Analog zeigt man $\nu = \nu'$ und $c_j = c'_j$. □

Lemma

SIMPLEX *terminiert nach höchstens* $\binom{n+m}{n}$ *Iterationen oder nie.*

Beweis: Anzahl der Möglichkeiten, S zu wählen, ist $\binom{n+m}{n}$. Wenn SIMPLEX mehr als $\binom{n+m}{n}$ Iterationen ausführt, muss er kreisen und terminiert daher nie. □

Regel von Bland: Wählt man immer eine Variable mit kleinstmöglichem Index als x_k und x_g, so terminiert SIMPLEX immer.

Satz

Falls INIT-SIMPLEX *eine zulässige Basislösung ausgibt und die Regel von Bland eingehalten wird, dann gilt für den* SIMPLEX*-Algorithmus:*

- *Das lineare Programm ist unbeschränkt, oder*
- *man erhält die korrekte Lösung nach höchstens* $\binom{n+m}{n}$ *Iterationen.*

Suchen der initialen zulässigen Lösung

Beispiel:

$$\begin{aligned} 2x_1 - x_2 \quad & \text{max!} \\ 2x_1 - 2x_2 &\leq 2 \\ x_1 - 5x_2 &\leq -3 \\ x_1, x_2 &\geq 0 \end{aligned}$$

Schlupfform:

$$\begin{aligned} z &= 2x_1 - x_2 \\ x_3 &= 2 - 2x_1 + 2x_2 \\ x_4 &= -3 - x_1 + 5x_2 \end{aligned}$$

Basislösung $(0, 0, 2, -3)$ nicht zulässig!

$\Longrightarrow$ INIT-SIMPLEX kann nicht einfach offensichtliche Schlupfform ausgeben.

Lineare Programmierung

$\longrightarrow$ lineares Hilfsprogramm:

- stellt Lösbarkeit fest;
- findet Schlupfform mit zulässiger Basislösung.

Lemma

Sei L ein lineares Programm in Standardform $(A, \mathbf{b}, \mathbf{c})$. Wir definieren das Programm L_H durch

$$\begin{aligned} -x_0 \quad & \text{max!} \\ \sum_{j=1}^{n} a_{i,j} x_j - x_0 &\leq b_i, && i = 1, \ldots, m; \\ x_j &\geq 0, && j = 0, 1, \ldots, n. \end{aligned}$$

Dann ist L genau dann lösbar, wenn $x_0 = 0$ der optimale Zielfunktionswert von L_H ist.

Beweis: „$\Longrightarrow$“: Sei $\mathbf{x} = (x_1, \ldots, x_n)$ eine Lösung von L.
Dann ist $(0, x_1, \ldots, x_n)$ eine Lösung von L_H.
Wegen VS $x_0 \geq 0$ ist $-x_0 > 0$ unmöglich $\Longrightarrow$ $x_0 = 0$ optimal.

„$\Longleftarrow$“: Es existiere eine Lösung $\mathbf{x}$ von L_H mit $x_0 = 0$.
Dann löst $(x_1, \ldots, x_n)$ das Programm L. □

Algorithm INIT-SIMPLEX$(A, \mathbf{b}, \mathbf{c})$

1: Sei k der Index von $b_k = \min_i b_i$
2: **if** $b_k \geq 0$ **then return** $(\underbrace{\{1, 2, \ldots, n\}}_{F}, \underbrace{\{n+1, n+2, \ldots, n+m\}}_{S}, A, \mathbf{b}, \mathbf{c}, \underbrace{0}_{\nu})$
3: **end if**
4: Bilde L_H durch Addition von $-x_0$ zur linken Seite aller Nebenbedingungen and Setzen von $z = -x_0$.
5: Sei $(F, S, A, \mathbf{b}, \mathbf{c}, \nu)$ die dadurch entstehende Schlupfform.
6: $g := n + k$
7: $(F, S, A, \mathbf{b}, \mathbf{c}, \nu) :=BW(F, S, A, \mathbf{b}, \mathbf{c}, \nu, g, 0)$
8: Bestimme eine optimale Lösung z von L_H durch Iterieren der **while**-Schleife von SIMPLEX.
9: **if** $z = 0$ **then**
10: **if** $0 \in S$ **then** Basiswechsel, um $0 \in F$ zu erzwingen.
11: **end if**
12: Entferne alle x_0-Terme aus der Schlupfform von L_H.
13: Ersetze z durch die Zielfunktion von L.
14: Ersetze alle Schlupfvariablen durch die rechte Seite der ihr entsprechenden Nebenbedingung.
15: **return** die so gewonnene Schlupfform
16: **else return** „unlösbar“
17: **end if**

Beispiel zu INIT-SIMPLEX

$$\begin{aligned} 2x_1 - x_2 \quad & \text{max!} \\ 2x_1 - 2x_2 &\leq 2 \\ x_1 - 5x_2 &\leq -3 \\ x_1, x_2 &\geq 0 \end{aligned}$$

Hilfsprogramm:

$$\begin{aligned} -x_0 \quad & \text{max!} \\ 2x_1 - 2x_2 - x_0 &\leq 2 \\ x_1 - 5x_2 - x_0 &\leq -3 \\ x_0, x_1, x_2 &\geq 0 \end{aligned}$$

Hilfsprogramm in Schlupfform:

$$\begin{aligned} z &= -x_0 \\ x_3 &= 2 - 2x_1 + 2x_2 + x_0 \\ x_4 &= -3 - x_1 + 5x_2 + x_0 \end{aligned}$$

Basiswechsel mit $x_0 \leftrightarrow x_4$ liefert

$$\begin{aligned} z &= -3 - x_1 + 5x_2 - x_4 \\ x_0 &= 3 + x_1 - 5x_2 + x_4 \\ x_3 &= 5 - x_1 - 3x_2 + x_4 \end{aligned}$$

Basislösung ist $\mathbf{x} = (3, 0, 0, 5, 0)$: zulässig!

Weiterer Basiswechsel mit $x_0 \leftrightarrow x_2$ liefert

$$\begin{aligned} z &= -x_0 \\ x_2 &= \frac{3}{5} - \frac{1}{5}x_0 + \frac{1}{5}x_1 + \frac{1}{5}x_4 \\ x_3 &= \frac{16}{5} + \frac{3}{5}x_0 - \frac{8}{5}x_1 + \frac{2}{5}x_4 \end{aligned}$$

mit zulässiger Basislösung $\mathbf{x}' = (0, 0, \frac{3}{5}, \frac{16}{5}, 0)$.

Da $x_0 = 0$ ist $\mathbf{x}'$ optimal und L lösbar!

Lemma

Falls das lineare Programm L unlösbar ist, dann gibt INIT-SIMPLEX *„unlösbar“aus, andernfalls eine Schlupfform mit zulässiger Basislösung.*

Beweis: 1. Fall: L unlösbar

Das Hilfsprogramm L_H hat optimalen Zielfunktionswert $z_0 < 0$.

Setzen wir

$$x_i = 0 \text{ für } i = 1, \ldots, n \text{ und } x_0 = |\min_{1 \leq i \leq m} b_i|,$$

so erhalten wir eine zulässige Lösung mit

$$z = -|\min_{1 \leq i \leq m} b_i|,$$

also gilt $z_0 > -\infty$.

Daher findet INIT-SIMPLEX eine Lösung mit $z < 0$.

Für die Basislösung der letzten Schlupfform gilt $x_0 > 0$, da $z_0 < 0$.

Somit gibt INIT-SIMPLEX „unlösbar “aus.

2. Fall: L besitzt zulässige Lösung

Falls alle $b_i \geq 0$, so ist die Basislösung zulässig. Dann wird die Eingabe in Schlupfform ausgegeben.

Falls es ein i mit $b_i < 0$ gibt, dann ist $b_k \geq b_i < 0$.
Nun erfolgt ein Basiswechsel mit $x_0 \leftrightarrow x_{n+k}$; die zugehörige Nebenbedingung lautet

$$x_{n+k} = \underbrace{b_k}_{<0} - \sum_{j=1}^{n} a_{k,j} x_j + x_0.$$

Zu zeigen: Nach diesem Basiswechsel sind alle $\hat{b}_i$ nichtnegativ.

Sei $\bar{\mathbf{x}}$ die Basislösung nach diesem Basiswechsel. Dann gilt (voriges Lemma!)

$$\bar{x}_i = \begin{cases} b_i - a_{i,0}\hat{b}_0, & \text{falls } i \in \hat{S} \setminus \{0\}; \\ b_k / a_{k,0}, & \text{falls } i = 0 \end{cases}$$

Beachte:

$$\sum_{j=1}^{n} a_{i,j}x_j - x_0 \leq b_i \quad \Longleftrightarrow \quad \sum_{j=0}^{n} a_{i,j}x_j \leq b_i,$$

also $a_{i,0} = -1$.

Wegen $b_k < 0$ und $a_{k,0} = -1$ gilt daher $b_k/a_{k,0} > 0$.
Weiters haben wir $\bar{x}_k = \bar{x}_0$ und $\bar{x}_g = \bar{x}_{n+k}$ (k-te Schlupfvariable!).
Die erste Zuweisung in BW lautet daher

$$\hat{b}_0 := \frac{b_{n+k}}{a_{n+k,0}},$$

wobei b_{n+k} bzw. $a_{n+k,0}$ der Schlupfform genau die Größen b_k bzw. $a_{k,0}$ des Ausgangsprogramms $(A, \mathbf{b}, \mathbf{c})$ in Standardform sind.

Daher erhalten wir

$$b_i - a_{i,0}\hat{b}_0 = b_i - a_{i,0}\frac{b_k}{a_{k,0}} = b_i - b_k \geq 0.$$

Die Basislösung ist also zulässig, L lösbar und die optimale Lösung von L_H erfüllt daher $z_0 = 0$.

Wegen der Äquivalenz der neuen und der alten Schlupfform haben wir $\bar{x}_0 = 0$.
Entfernen von x_0 im linearen Programm
$\rightsquigarrow$ Schlupfform von L mit zulässiger Basislösung, die ausgegeben wird. □

Satz (Fundamentalsatz der linearen Programmierung)

Sei L ein lineares Programm in Standardform. Dann gilt genau eine der folgenden Aussagen:

- *L besitzt eine optimale Lösung mit $z_0 < \infty$.*
- *L ist unlösbar.*
- *L ist unbeschränkt*

In den jeweiligen Fällen gibt SIMPLEX

- *eine optimale Lösung mit endlichem z,*
- *„unlösbar“ bzw.*
- *„unbeschränkt“*

aus.