

Lineare Programmierung

Grundidee eines linearen Programmierungsproblems:

Eine lineare Funktion mit n Variablen maximieren oder minimieren, wobei für die Variablenbelegung bestimmte Einschränkungen gelten.
(z.B. beschränkte Ressourcen oder konkurrierende Nebenbedingungen).

Ein lineares Programm besteht aus einer zu optimierenden linearen Funktion

$$f(x_1, x_2, \dots, x_n) = a_1 \cdot x_1 + a_2 \cdot x_2 + \dots + a_n \cdot x_n = \sum_{i=1}^n a_i x_i$$

unter bestimmten linearen Nebenbedingungen.

Sei b eine reelle Zahl und f eine **lineare Funktion**, dann ist

$f(x_1, x_2, \dots, x_n) = b$ eine **lineare Gleichung** und

$f(x_1, x_2, \dots, x_n) \geq b$ eine **lineare Ungleichung**

$f(x_1, x_2, \dots, x_n) \leq b$ eine **lineare Ungleichung**

Lineare Nebenbedingungen sind sowohl lineare Gleichungen als auch lineare Ungleichungen.

Begriffe:

Zulässige Lösung:

Variablenbelegung, die alle Nebenbedingungen erfüllt

Unzulässige Lösung:

Alle anderen Variablenbelegungen

Unlösbar:

Es gibt keine zulässige Lösung für ein lineares Programm

Zielfunktionswert:

Wert, den die Zielfunktion bei einer bestimmten Variablenbelegung annimmt

Optimale Lösung:

Zulässige Lösung mit optimalem, d.h. maximalem Zielfunktionswert

Unbeschränktes Programm:

Zulässige Lösungen existieren, aber keine optimale Lösung

Äquivalenz linearer Programme:

Zwei lineare Maximierungs-Programme heißen **äquivalent**, wenn die jeweilige Menge der zulässigen Zielfunktionswerte äquivalent ist.

Ein lineares Maximierungs-Programm L und ein lineares

Minimierungsprogramm L' heißen **äquivalent**, wenn jedem z aus der Menge der zulässigen Zielfunktionswerte von L ein $-z$ aus der Menge der zulässigen Zielfunktionswerte von L' entspricht

Es gibt zwei verschiedene Darstellungen von linearen Programmen:

Die **Standardform** und die **Schlupfform** (auch Slackform).

Standardform:

Ein lineares Programm in Standardform ist die Maximierung einer linearen

Funktion $\sum_{j=1}^n c_j \cdot x_j$ (Zielfunktion)

unter Beachtung linearer Ungleichungen $\sum_{j=1}^n a_{ij} \cdot x_j \leq b_i, 1 \leq i \leq m$

(Nebenbedingungen)

und $x_j \geq 0$ für $j = 1, 2, \dots, n$ (Nichtnegativitätsbedingungen)

Man kann jedes lineare Programm in die Standardform bringen:

Es gibt 4 Gründe, warum ein lineares Programm nicht in Standardform ist

- Zielfunktion soll minimiert anstatt maximiert werden
- Es gibt Variablen ohne Nichtnegativitätsbedingung
- Es gibt Nebenbedingungen, die eine Gleichung sind
- Es gibt Nebenbedingungen, die eine Ungleichung sind, die aber „ \geq “ statt „ \leq “ fordern

Lösung:

- Minimierung \Rightarrow Maximierung
Negiere alle Koeffizienten der Zielfunktion
- Nichtnegativitätsbedingung erfüllen
Führe für x_i zwei neue Variablen $x_i' \geq 0$ und $x_i'' \geq 0$ ein und substituier alle x_i mit $x_i' - x_i''$
- Gleichung \Rightarrow Ungleichung
Ersetze $A = B$ durch $A \leq B$ und $A \geq B$
- \geq -Relation \Rightarrow \leq -Relation
Multipliziere beide Seiten mit (-1)

Bsp. Für ein lineares Programm in Standardform:

maximiere $2x_1 - 3x_2 + 3x_3$

unter den Nebenbedingungen

$$x_1 + x_2 - x_3 \leq 7$$

$$-x_1 - x_2 + x_3 \leq -7$$

$$x_1 - 2x_2 + 2x_3 \leq 4$$

$$x_1, x_2, x_3 \geq 0$$

Schlupfform:

Ein lineares Programm in Schlupfform ist die Maximierung einer linearen Funktion unter Beachtung linearer Gleichungen.

Man kann jedes Programm in Schlupfform bringen.

Zuerst bringt man das Programm in Standardform. Dann müssen nur noch Ungleichungen in Gleichungen umgeformt werden.

$$\sum_{j=1}^n a_{ij} x_j \leq b_i$$

wird umgeformt zu

$$s = b_i - \sum_{j=1}^n a_{ij} x_j$$

s ist die Schlupfvariable. Sie ist ein Maß für die Differenz(oder den Schlupf) zwischen der linken und der rechten Seite der Ungleichung.
 Wenn wir ein Programm aus der Standard in die Schlupfform bringen, bezeichnen wir die Schlupfvariablen nicht mit s sondern mit x_{n+1} .
 Die i-te Nebenbedingung wäre also:

$$x_{n+i} = b_i - \sum_{j=1}^n a_{ij}x_j$$

$$x_{n+i} \geq 0$$

Bsp. Für ein Programm in Schlupfform:

(Man schreibt die Zielfunktion als Gleichung und erwähnt nicht explizit die Maximierung. Auch die Nichtnegativitätsbedingungen spart man sich)

$$z = 2x_1 - 3x_2 + 3x_3$$

$$x_4 = 7 - x_1 - x_2 + x_3$$

$$x_5 = -7 + x_1 + x_2 - x_3$$

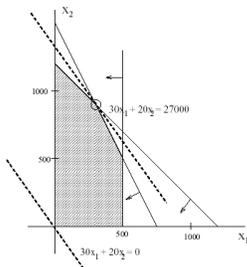
$$x_6 = 4 - x_1 + 2x_2 - 2x_3$$

Auf der rechten Seite stehen jeweils die gleichen Variablen, man nennt sie die **Nichtbasisvariablen**.

Auf der linken Seite stehen die **Basisvariablen**.

Graphische Interpretation:

- Variablenbelegung $x = (x_1, x_2, \dots, x_n)$ entspricht Punkt im n-dim. Raum
- Nebenbedingungen sind Halbräume im n-dim. Raum (Die Grenzen der Halbräume sind Hyperebene)
- Die Schnittmenge aller Halbräume stellt den Bereich aller zulässiger Lösungen (**zulässiger Bereich**)



Lemma: (Konvexität des zulässigen Bereichs)

Die Punktmenge, die durch das Polyeder P beschrieben wird, ist konvex.
 D.h. für alle Punkte $x \in P$ und $y \in P$ gilt, dass die Punkte, die auf der Strecke zwischen x und y liegen auch in P enthalten sind.

Lemma:

Ein lokales Optimum des Polyeders ist immer gleichzeitig auch ein globales Optimum.

Lemma:

Wenn es für ein lineares Programm eine optimale Lösung gibt, so gibt es eine Ecke des zugehörigen Polyeders, deren Zielfunktionswert optimal ist.

Der Simplexalgorithmus

Er ist das klassische Verfahren zur Lösung linearer Programme.

Seine Laufzeit ist im schlechtesten Fall nicht polynomiell, doch meist ist er in der Praxis sehr schnell.

Er bekommt als Eingabe ein lineares Programm in Schlupfform und gibt eine optimale Lösung aus.

Er beginnt an einer Ecke des Simplex (dies ist am Anfang der Ursprung) und bewegt sich dann in jedem Schritt von einer Ecke zur nächsten, deren Zielfunktionswert nicht kleiner ist (und meist ist er größer).

Der Simplexalgorithmus terminiert, wenn er ein lokales Maximum gefunden hat. Da der zulässige Bereich konvex ist, ist dies dann gleichzeitig ein globales Maximum.

Wie man erkennen kann erspart der Simplexalgorithmus zwar gewöhnlich den Weg durch alle Ecken, doch es können durchaus auch mal sehr viele Ecken zwischen der Startecke und der optimalen Ecke abgelaufen werden.

Grundidee, die hinter jedem Schritt (jeder Iteration) des Algorithmus steckt:

Jeder Iteration des Programms ist eine Basislösung zugeordnet, man setzt alle Nichtbasisvariablen auf 0 und berechnet die Werte der Basisvariablen aus den Gleichungsnebenbedingungen.

Eine Basislösung entspricht immer einer Ecke des Simplex.

Bei der Iteration formt man die Schlupfform in eine äquivalent andere Schlupfform um.

Damit der Zielfunktionswert bei der Umformung nicht kleiner wird, wählen wir eine Nichtbasisvariable so aus, dass der Zielfunktionswert grösser wird, wenn wir diese Variable erhöhen.

Danach formen wir das Programm so um, das aus der Nichtbasisvariable eine Basisvariable wird.

Lemma:

Falls der Simplexalgorithmus nach maximal $\binom{n+m}{m}$ Iterationen nicht terminiert,

dann kreiselt er.

Andere Verfahren zum Lösen von linearen Programmen sind das Ellipsoid und das Punkt-in-Punkt-Verfahren. Dies sind Algorithmen, die in polynomieller Zeit laufen.