

Lineare Optimierung

Vollstaendige Vorlesung

BSc Analysis

- 1 Einführung
- 2 Historischer Kontext
- 3 Grundlagen
- 4 Erweiterte Grundlagen
- 5 Graphische Lösung
- 6 Sonderfaelle ausfuehrlich
- 7 Simplex-Verfahren
- 8 Big-M-Methode
- 9 Zweiphasen-Methode
- 10 Dualitaet

Am Ende dieser Lektion koennen Sie:

1. Ein **lineares Programm** in Standardform formulieren und die Bestandteile benennen.
2. Den **zulaessigen Bereich** geometrisch interpretieren und den **Eckpunktsatz** beweisen.
3. Ein 2D-LP **graphisch loesen** und das Optimum an Eckpunkten bestimmen.
4. Das **Simplex-Verfahren** tabellarisch durchfuehren und die Optimalloesung ablesen.
5. **Big-M-** und **Zweiphasen-Methode** anwenden, wenn der Ursprung nicht zulaessig ist.
6. Den **dualen LP** aufstellen, den starken Dualitaetsatz und komplementaeren Schlupf anwenden.
7. **Sensitivitaetsanalyse** durchfuehren und die Ergebnisse oekonomisch interpretieren.
8. Grundideen von **ganzzahliger Optimierung** und **Inneren-Punkte-Methoden** beschreiben.

Lineare Optimierung ist eines der am haeufigsten eingesetzten Verfahren in Wirtschaft, Logistik und Ingenieurwesen.

Meilensteine:

- **1939 – Kantorovich:** Formuliert LP-Probleme fuer Produktionsplanung in der Sowjetunion. Nobelpreis 1975.
- **1947 – Dantzig:** Entwickelt den **Simplex-Algorithmus** fuer die US Air Force (Logistikplanung).
- **1947 – von Neumann:** Formuliert die Dualitaetstheorie.
- **1958 – Gomory:** Schnittebenen fuer ganzzahlige Optimierung.
- **1979 – Khachiyan:** Ellipsoid-Methode – erster **polynomialer** LP-Algorithmus.
- **1984 – Karmarkar: Innere-Punkte-Methode** – polynomial und praktisch schnell.

Bedeutung heute:

- LP wird in nahezu jeder Industrie eingesetzt
- Flugplaene, Lieferketten, Energienetze
- Milliarden von LPs werden taeglich geloest
- Basis fuer Machine Learning (SVMs, L_1 -Regression)

Komplexitaetstheorie:

- Simplex: Worst Case **exponentiell** (Klee-Minty 1972), Praxis: sehr schnell
- Ellipsoid (Khachiyan): **polynomial**, aber langsam
- Interior Point (Karmarkar): **polynomial** und schnell
- $LP \in P$ (polynomiale Klasse)

Die LP-Theorie vereint Analysis, Algebra und Geometrie – und hat tiefe Verbindungen zur Spieltheorie und Oekonomie.

Operations Research (OR):

Definition

Wissenschaftliche Methoden zur Entscheidungsunterstuetzung bei komplexen Planungs- und Steuerungsproblemen.

LP als Fundament von OR:

- Ca. 70% aller OR-Modelle enthalten ein LP als Teilproblem
- LP ist "Arbeitspferd" der Optimierung
- Schnelle Solver ermöglichen Echtzeitentscheidungen

Einsatzgebiete:

- Logistik & Supply Chain
- Finanzplanung & Portfolio
- Energiewirtschaft & Netze
- Personalplanung & Scheduling

Dantzig nannte LP das "most successful mathematical model" – eine Einschätzung, die bis heute gilt.

Groessenordnungen realer LPs:

Anwendung	Var.	NB
Baeckerei (Lehre)	2	2
Produktionsplan	10^2	10^2
Transportnetz	10^4	10^3
Airline Scheduling	10^6	10^5
Strommarkt	10^7	10^6

Solver-Performance:

- Gurobi/CPLEX: Millionen Variablen in Sekunden
- HiGHS (Open Source): sehr konkurrenzfaehig
- 100-fache Beschleunigung in 25 Jahren (1990–2015)

Definition

Ein **lineares Programm (LP)** besteht aus:

- einer **linearen Zielfunktion**, die maximiert oder minimiert wird,
- **linearen Nebenbedingungen** (Gleichungen oder Ungleichungen),
- **Nichtnegativtaetsbedingungen** fuer die Entscheidungsvariablen.

Allgemeine Form:

$$\begin{aligned} & \underset{\mathbf{x}}{\text{maximize}} \quad \mathbf{c}^T \mathbf{x} \\ \text{s.t.} \quad & \mathbf{Ax} \leq \mathbf{b}, \quad \mathbf{x} \geq \mathbf{0} \end{aligned}$$

Dabei:

- $\mathbf{x} \in \mathbb{R}^n$: Entscheidungsvariablen
- $\mathbf{c} \in \mathbb{R}^n$: Zielfunktionskoeffizienten
- $\mathbf{A} \in \mathbb{R}^{m \times n}$: Koeffizientenmatrix
- $\mathbf{b} \in \mathbb{R}^m$: rechte Seite

Linearitaet: Zielfunktion und Nebenbedingungen sind linear in den Entscheidungsvariablen – keine Produkte $x_1 x_2$ oder Potenzen x_1^2 .

Beispiel (Baeckerei):

Eine Baeckerei stellt Brot (x_1) und Kuchen (x_2) her.

- Gewinn: $3x_1 + 5x_2$ Euro
- Mehl: $x_1 + 2x_2 \leq 8$ kg
- Arbeitszeit: $3x_1 + 2x_2 \leq 12$ Std.
- $x_1, x_2 \geq 0$

⇒ **Wie viel von jedem Produkt maximiert den Gewinn?**

Standardform (Maximierung)

$$\underset{\mathbf{x} \in \mathbb{R}^n}{\text{maximize}} \quad z = \mathbf{c}^T \mathbf{x} \quad \text{s.t.} \quad \mathbf{A} \mathbf{x} \leq \mathbf{b}, \quad \mathbf{x} \geq \mathbf{0}$$

Unser Baeckerei-Beispiel:

$$\begin{aligned} \underset{x_1, x_2}{\text{maximize}} \quad & z = 3x_1 + 5x_2 \\ \text{s.t.} \quad & x_1 + 2x_2 \leq 8 \\ & 3x_1 + 2x_2 \leq 12 \\ & x_1, x_2 \geq 0 \end{aligned}$$

In Matrixform:

$$\mathbf{c} = \begin{pmatrix} 3 \\ 5 \end{pmatrix}, \quad \mathbf{A} = \begin{pmatrix} 1 & 2 \\ 3 & 2 \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} 8 \\ 12 \end{pmatrix}$$

Umwandlungsregeln:

- Minimierung \rightarrow Maximierung:
 $\text{minimize } f(\mathbf{x}) = \text{maximize } (-f(\mathbf{x}))$
- \geq -Nebenbedingung: mit (-1) multiplizieren
- Gleichung $= \rightarrow$ zwei Ungleichungen (\leq und \geq)
- Variable ohne Vorzeichen-Beschränkung:
 $x_j = x_j^+ - x_j^-$ mit $x_j^+, x_j^- \geq 0$

Merke: Jedes LP lässt sich in Standardform bringen.

Die Standardform ist Voraussetzung fuer das Simplex-Verfahren – alle Algorithmen erwarten dieses Format.

Definition

Der **zulaessige Bereich** ist die Menge aller \mathbf{x} , die alle Nebenbedingungen erfullen:

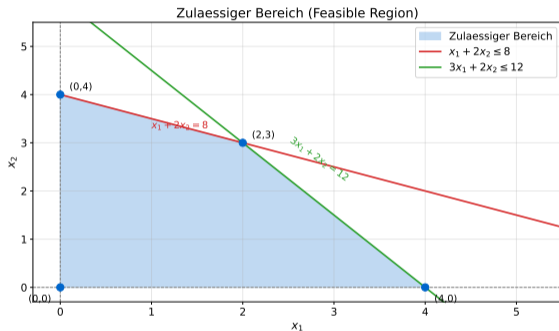
$$\mathcal{F} = \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{Ax} \leq \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$$

Eigenschaften:

- \mathcal{F} ist ein **konvexes Polyeder**
- Schnittmenge endlich vieler Halbraeume
- Jede Nebenbedingung definiert einen Halbraum
- Die Nichtnegativitaet beschraenkt auf den 1. Quadranten

Eckpunkte: $(0,0)$, $(4,0)$, $(2,3)$, $(0,4)$.

Konvexitaet ist entscheidend: Jede Verbindungslinie zwischen zwei zulaessigen Punkten liegt komplett im zulaessigen Bereich.



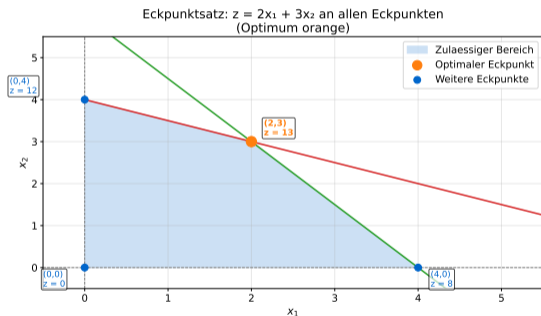
Satz (Eckpunktsatz)

Besitzt ein LP eine optimale Lösung, so gibt es eine **optimale Lösung an einem Eckpunkt** des zulaessigen Bereichs.

Beweisskizze:

1. Sei \mathbf{x}^* optimal und *kein* Eckpunkt.
2. Dann $\mathbf{x}^* = \lambda \mathbf{u} + (1 - \lambda) \mathbf{v}$ fuer zulaessige \mathbf{u}, \mathbf{v} und $0 < \lambda < 1$.
3. Da z linear: $z(\mathbf{x}^*) = \lambda z(\mathbf{u}) + (1 - \lambda) z(\mathbf{v})$.
4. Mindestens einer von $z(\mathbf{u}), z(\mathbf{v})$ ist $\geq z(\mathbf{x}^*)$.
5. Wiederholung fuehrt zu einem Eckpunkt.

Konsequenz: Endlich viele Eckpunkte pruefen genuegt!



Auswertung:

Ecke	x_1	x_2	z
------	-------	-------	-----

Baekerei-LP: maximize $z = 3x_1 + 5x_2$ unter den bekannten Nebenbedingungen.

Berechnung von $C = (2, 3)$:

$$x_1 + 2x_2 = 8$$

$$3x_1 + 2x_2 = 12$$

Subtraktion: $2x_1 = 4 \Rightarrow x_1 = 2, x_2 = 3$.

Verifikation:

- $2 + 6 = 8 \leq 8$ ✓
- $6 + 6 = 12 \leq 12$ ✓

Ergebnis

Optimalloesung: $x_1^* = 2, x_2^* = 3$

Optimaler Gewinn: $z^* = 21$ Euro

\Rightarrow 2 Einheiten Brot, 3 Einheiten Kuchen.

Oekonomische Deutung:

- Beide Ressourcen voll ausgeschopft (Schlupf = 0)
- Kein Spielraum fuer Produktionserhoehung
- Frage: Was waere 1 kg Mehl wert? \rightarrow Dualitaet

Bei kleinen LPs mit 2 Variablen genuegt es, alle Eckpunkte zu berechnen und die Zielfunktion auszuwerten.

Standardform (Ungleichungen)

$$\begin{aligned} &\text{maximize} && \mathbf{c}^T \mathbf{x} \\ &\text{s.t.} && \mathbf{Ax} \leq \mathbf{b}, \quad \mathbf{x} \geq \mathbf{0} \end{aligned}$$

Kanonische Form (Gleichungen)

$$\begin{aligned} &\text{maximize} && \mathbf{c}^T \mathbf{x} \\ &\text{s.t.} && \mathbf{Ax} = \mathbf{b}, \quad \mathbf{x} \geq \mathbf{0} \end{aligned}$$

Umwandlung: $\mathbf{Ax} \leq \mathbf{b} \xrightarrow{+\mathbf{s}} \mathbf{Ax} + \mathbf{Is} = \mathbf{b}, \mathbf{s} \geq \mathbf{0}$. Erweiterter Vektor: $\tilde{\mathbf{x}} = (\mathbf{x}^T, \mathbf{s}^T)^T \in \mathbb{R}^{n+m}$.

Beide Formen sind äquivalent – die Wahl hängt vom Algorithmus ab.

Verschiedene NB-Typen in die Standardform:

1. **\geq -Ungleichung:** $2x_1 + x_2 \geq 6 \xrightarrow{\times(-1)} -2x_1 - x_2 \leq -6$
2. **Gleichung:** $x_1 + x_2 = 5 \rightarrow x_1 + x_2 \leq 5$ und $-x_1 - x_2 \leq -5$
3. **Minimierung:** minimize $2x_1 - 3x_2 \rightarrow$ maximize $-2x_1 + 3x_2$
4. **Freie Variable:** $x_j = x_j^+ - x_j^-$, $x_j^+, x_j^- \geq 0$

Zusammenfassung

Jedes LP kann durch systematische Anwendung dieser Regeln in die Standardform gebracht werden.

In der Praxis uebernehmen Solver diese Umwandlungen automatisch (Preprocessing).

Beispiel mit 3 Produkten:

$$\begin{aligned} \text{maximize} \quad & z = 3x_1 + 5x_2 + 2x_3 \\ \text{s.t.} \quad & x_1 + 2x_2 + x_3 \leq 10 \\ & 3x_1 + x_2 + 2x_3 \leq 12 \\ & x_1 + x_2 + x_3 \leq 8 \\ & x_1, x_2, x_3 \geq 0 \end{aligned}$$

Graphische Loesung nicht mehr moeglich.

Anzahl moeglicher Eckpunkte:

n	m	Max. Eckpunkte
2	2	6
3	3	20
5	5	252
10	10	184 756

⇒ Simplex waehlt gezielt vielversprechende Eckpunkte, statt alle abzusuchen.

Reale LPs haben oft Tausende von Variablen – nur intelligente Algorithmen sind hier machbar.

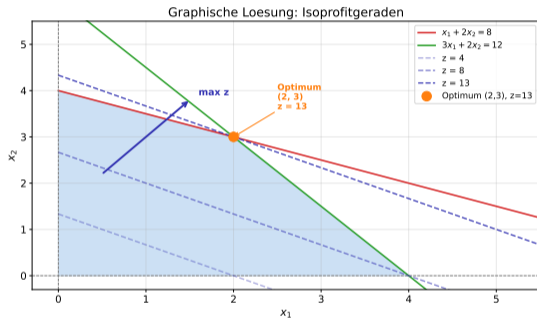
Idee

Isoprofit-Linie: $\mathbf{c}^T \mathbf{x} = k$, d. h. $3x_1 + 5x_2 = k$.

Vorgehen:

1. Zeichne den zulaessigen Bereich.
2. Zeichne Isoprofit-Linie fuer beliebiges k .
3. Verschiebe parallel in Richtung $\nabla z = \mathbf{c}$.
4. Letzter Beruehrpunkt mit \mathcal{F} ist das Optimum.

Gradient: $\nabla z = \mathbf{c} = (3, 5)^T$ zeigt in Richtung wachsender z -Werte.



Die graphische Methode funktioniert nur fuer 2 (maximal 3) Variablen – fuer hoehere Dimensionen brauchen wir den Simplex.

Schritt 1: Nebenbedingungen als Geraden.

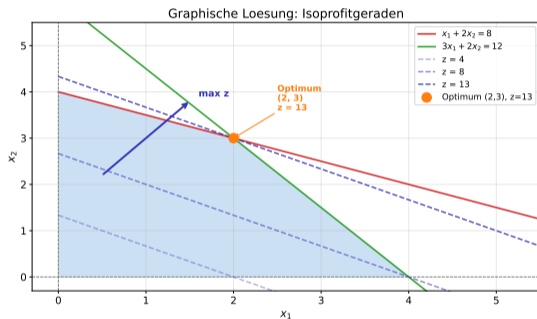
- $x_1 + 2x_2 = 8$: Abschnitte (8, 0) und (0, 4)
- $3x_1 + 2x_2 = 12$: Abschnitte (4, 0) und (0, 6)

Schritt 2: Isoprofit-Linien.

- $z = 5, 15, 21$ (parallel verschieben)

Schritt 3: Optimum ablesen.

- Letzter Kontakt bei (2, 3), $z^* = 21$.



Die parallele Verschiebung der Isoprofit-Linie sichert, dass das globale Optimum gefunden wird.

Verifikation der Loesung $(x_1^*, x_2^*) = (2, 3)$:

1. **Zulaessigkeit:** $x_1 + 2x_2 = 8 \leq 8 \checkmark$, $3x_1 + 2x_2 = 12 \leq 12 \checkmark$
2. **Optimalitaet:** Beide NB aktiv (Schnittpunkt).
3. **Zielfunktionswert:** $z^* = 21$.

Oekonomische Deutung:

- 2 Einheiten Brot, 3 Einheiten Kuchen
- Gesamtgewinn: 21 Euro
- Schlupf: $s_1 = s_2 = 0$

Beobachtung

Beide Ressourcen vollstaendig ausgeschoept: Mehl und Arbeitszeit sind knapp (binding constraints).

Aktive Nebenbedingungen zeigen, welche Ressourcen den Gewinn limitieren – das fuehrt zur Dualitaet.

Unbeschraenktes LP

z kann beliebig gross werden. Keine endliche Optimalloesung.

Mehrdeutige Loesung

Unendlich viele Optima entlang einer Kante. Isoprofit-Linie parallel zu aktiver NB.

Unzulaessiges LP

$\mathcal{F} = \emptyset$, keine Loesung existiert.

Degeneriertes LP

Eckpunkt im Schnitt von mehr als n aktiven NB. Kann zu Zyklen fuehren.

Unbeschraenkte und unzulaessige LPs sind meist Zeichen eines Modellierungsfehlers.

Erkennung

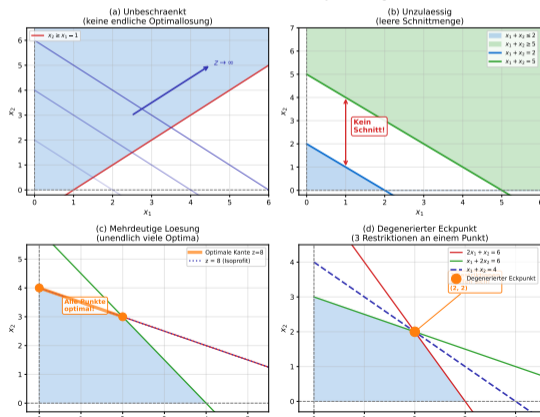
Wenn in der Pivotspalte **kein positiver Eintrag** existiert: $a_{ij} \leq 0$ fuer alle $i \Rightarrow z \rightarrow \infty$.

Beispiel: maximize $2x_1 + x_2$ mit $-x_1 + x_2 \leq 4$,
 $x_1, x_2 \geq 0$.

Pivotspalte x_1 : Eintrag = $-1 < 0$ – kein positiver Quotient moeglich!

Geometrisch: Zulaessiger Bereich nach rechts offen.

Sonderfaelle der Linearen Optimierung



Beispiel: $x_1 + x_2 \leq 4$ und $x_1 + x_2 \geq 6$ gleichzeitig $\Rightarrow \mathcal{F} = \emptyset$.

Im Simplex: Kuenstliche Variablen koennen nicht auf 0 gedruickt werden.

Diagnose:

1. NB paarweise auf Widersprueche pruefen.
2. Relaxierung: NB einzeln entfernen.
3. Solver: IIS-Analyse (Irreducible Infeasible Subsystem).

Haeufige Ursachen:

- Widerspruechliche Kapazitaetsgrenzen
- Mindestmengen $\hat{}$ Kapazitaet
- Falsche Vorzeichen

Merke: Unzulaessigkeit ist ein Modellierungsproblem, kein algorithmisches Problem.

Solver wie Gurobi liefern das IIS – die minimale Menge widerspruechlicher Nebenbedingungen.

Definition

Basisloesung ist **degeneriert**, wenn mindestens eine Basisvariable = 0 (RHS-Eintrag = 0).

Konsequenzen:

- Pivotschritt kann z nicht verbessern (Seitensprung)
- Theoretisch: Zykeln moeglich (extrem selten)

Gegenstrategien:

- **Bland-Regel:** Kleinster Index \Rightarrow keine Zyklen
- **Perturbation:** Kleine Stoerung
- **Lexikographische Regel**

Beispiel: Am Eckpunkt $(2, 3)$ mit 3 aktiven NB bei $n = 2$: Ein Eckpunkt hat "zu viele" aktive NB.

Geometrisch: Mehr Hyperebenen schneiden sich in einem Punkt als zur Bestimmung des Eckpunkts noetig waeren.

Degenerierung ist geometrisch ein ueberbestimmter Eckpunkt – algebraisch eine Basisvariable mit Wert Null.

Erkennung

Im Optimaltableau hat eine Nichtbasisvariable $\bar{c}_j = 0$.

Beispiel: maximize $2x_1 + 4x_2$ mit $x_1 + 2x_2 \leq 8$,
 $3x_1 + 2x_2 \leq 12$.
Isoprofit-Linie parallel zu NB $x_1 + 2x_2 = 8$.

Alle Punkte auf der Kante optimal:

$$\mathbf{x}^* = \lambda(0, 4) + (1 - \lambda)(2, 3), \quad 0 \leq \lambda \leq 1$$

mit $z = 16$ fuer alle.

Praxis-Vorteil: Der Entscheider hat Spielraum fuer sekundaere Kriterien.

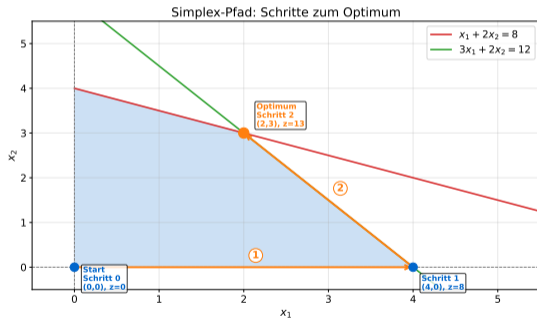
Alternative Optima sind kein Problem, sondern eine Chance: Sie bieten Flexibilitaet in der Entscheidung.

Idee

Starte an einem Eckpunkt und wandere entlang der Kanten zu benachbarten Eckpunkten mit hoeheren z -Werten, bis das Optimum erreicht ist.

Algorithmus:

1. Zulaessigen Eckpunkt waehlen (Start).
2. Optimalitaetstest.
3. Falls nein: Pivotspalte (verbessernde Richtung).
4. Pivotzeile (wie weit gehen?).
5. Basiswechsel (Pivotschritt).
6. Wiederhole ab 2.



Terminierung: Endlich viele Eckpunkte, jeder Schritt verbessert z (strikt) \Rightarrow kein Zykeln.

Umwandlung

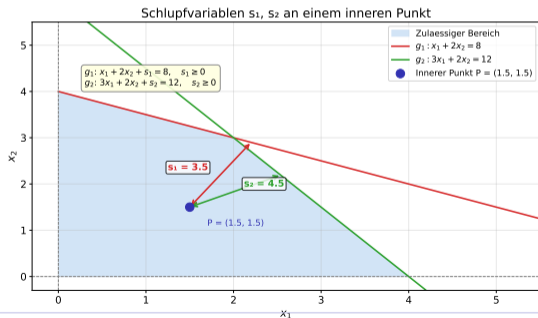
$$\mathbf{Ax} \leq \mathbf{b} \implies \mathbf{Ax} + \mathbf{s} = \mathbf{b}, \mathbf{s} \geq 0$$

Beispiel: $x_1 + 2x_2 + s_1 = 8, \quad 3x_1 + 2x_2 + s_2 = 12.$

Start: $x_1 = x_2 = 0, \quad s_1 = 8, \quad s_2 = 12, \quad z = 0$ (Ursprung).

Deutung:

- $s_i > 0$: Ressource i nicht ausgeschopft
- $s_i = 0$: NB i aktiv (binding)



Schlupfvariablen wandeln das LP in Gleichungsform um – Voraussetzung fuer die Simplex-Tabelle.

Ausgangstableau:

Basis	x_1	x_2	s_1	s_2	RHS
s_1	1	2	1	0	8
s_2	3	2	0	1	12
z	-3	-5	0	0	0

Leseweise:

- Zeilen: Nebenbedingungen
- Letzte Zeile: Zielfunktion (z-Zeile)
- RHS: rechte Seite

Basisloesung: $s_1 = 8$, $s_2 = 12$, $x_1 = x_2 = 0$, $z = 0$.

Die z-Zeile enthaelt die negativen Zielfunktionskoeffizienten: $z - 3x_1 - 5x_2 = 0$.

Pivotspalte: Spalte mit negativstem z -Eintrag (x_2 : -5).

Pivotzeile: Kleinster positiver Quotient: s_1

($8/2 = 4 < 12/2 = 6$).

	x_1	x_2	s_1	s_2	RHS	Quot.
s_1	1	2	1	0	8	4
s_2	3	2	0	1	12	6
z	-3	-5	0	0	0	

Die Quotientenregel (Minimum-Ratio-Test) stellt sicher, dass die neue Basisloesung zulaessig bleibt.

Gauss-Elimination: Pivotelement auf 1, Rest der Spalte auf 0.

Berechnung:

- Zeile 1: $\frac{1}{2}(1, 2, 1, 0 \mid 8) = (\frac{1}{2}, 1, \frac{1}{2}, 0 \mid 4)$
- Zeile 2: $(3, 2, 0, 1 \mid 12) - 2 \cdot Z'_1 = (2, 0, -1, 1 \mid 4)$
- z-Zeile: $(-3, -5, 0, 0 \mid 0) + 5 \cdot Z'_1 = (-\frac{1}{2}, 0, \frac{5}{2}, 0 \mid 20)$

Basis	x_1	x_2	s_1	s_2	RHS
x_2	$\frac{1}{2}$	1	$\frac{1}{2}$	0	4
s_2	2	0	-1	1	4
z	$-\frac{1}{2}$	0	$\frac{5}{2}$	0	20

$x_2 = 4, s_2 = 4, z = 20$. Noch $-\frac{1}{2} \Rightarrow$ nicht optimal.

Der Pivotschritt ist Gauss-Elimination – x_2 ersetzt s_1 in der Basis.

Zweiter Pivot: x_1 tritt ein ($-\frac{1}{2}$), s_2 tritt aus (Quotient $4/2 = 2$).

Basis	x_1	x_2	s_1	s_2	RHS
x_2	0	1	$\frac{3}{4}$	$-\frac{1}{4}$	3
x_1	1	0	$-\frac{1}{2}$	$\frac{1}{2}$	2
z	0	0	$\frac{9}{4}$	$\frac{1}{4}$	21

$x_1^* = 2$, $x_2^* = 3$, $z^* = 21$. Alle $\bar{c}_j \geq 0$ ✓ \Rightarrow **Optimum!**

Pfad: $(0, 0) \xrightarrow{z=0} (0, 4) \xrightarrow{z=20} (2, 3) \xrightarrow{z=21} \text{optimal}$.

Zwei Pivotschritte genuegen – der Simplex hat zwei Ecken besucht.

Kriterium

Optimal, wenn alle $\bar{c}_j \geq 0$ in der z-Zeile.

Interpretation:

- $\bar{c}_j < 0$: Erhoehung von x_j verbessert z
- $\bar{c}_j > 0$: Erhoehung verschlechtert z
- $\bar{c}_j = 0$: Alternative Optima moeglich

Endtableau: z-Zeile $(0, 0, \frac{9}{4}, \frac{1}{4} \mid 21)$.

Alle Eintraege ≥ 0 ✓.

Die Werte $\frac{9}{4}$ und $\frac{1}{4}$ unter s_1, s_2 sind die **Schattenpreise**.

Die reduzierten Kosten der Schlupfvariablen im Optimaltableau sind die Schattenpreise.

Zweites Beispiel: Ausgangstableau

LP: maximize $z = 4x_1 + 3x_2$ mit $2x_1 + x_2 \leq 10$, $x_1 + x_2 \leq 8$.

Basis	x_1	x_2	s_1	s_2	RHS	Quotient
s_1	2	1	1	0	10	5 ←
s_2	1	1	0	1	8	8
z	-4	-3	0	0	0	

Pivotspalte: x_1 (-4). Pivotzeile: s_1 ($5 < 8$).

Start: $s_1 = 10$, $s_2 = 8$, $x_1 = x_2 = 0$, $z = 0$.

Zweites Beispiel: Nach erstem Pivot

Basis	x_1	x_2	s_1	s_2	RHS	Quotient
x_1	1	$\frac{1}{2}$	$\frac{1}{2}$	0	5	10
s_2	0	$\frac{1}{2}$	$-\frac{1}{2}$	1	3	6 ←
z	0	-1	2	0	20	

$x_1 = 5, s_2 = 3, z = 20$. Noch -1 bei $x_2 \Rightarrow$ weiter pivotieren.

Pfad: $(0, 0) \rightarrow (5, 0)$ mit $z = 20$.

Basis	x_1	x_2	s_1	s_2	RHS
x_1	1	0	1	-1	2
x_2	0	1	-1	2	6
z	0	0	1	2	26

Optimalloesung

$x_1^* = 2$, $x_2^* = 6$, $z^* = 26$. Alle $\bar{c}_j \geq 0$ ✓.

Pfad: $(0, 0) \rightarrow (5, 0) \rightarrow (2, 6)$ optimal. Verifikation: $4 + 6 = 10 \leq 10$, $2 + 6 = 8 \leq 8$.

Schattenpreise: $y_1^* = 1$ (NB 1), $y_2^* = 2$ (NB 2).

Problem

Bei \geq -NB oder $=$ -NB liefern Schlupfvariablen keine zulaessige Startbasis. Der Ursprung ist nicht zulaessig!

Beispiel: minimize $2x_1 + 3x_2$ mit $x_1 + x_2 \geq 4$,

$2x_1 + x_2 \leq 10$.

$x_1 + x_2 - s_1 = 4$: Bei $x_1 = x_2 = 0$ wird $s_1 = -4 < 0$ ✗.

Loesung:

1. Kuenstliche Variable $a_1 \geq 0$ einfuegen.
2. $x_1 + x_2 - s_1 + a_1 = 4$.
3. Bestrafe: minimize $2x_1 + 3x_2 + Ma_1$.
4. Im Optimum: $a_1 = 0$.

M muss gross genug sein – typisch $M = 10^4$ bis 10^6 .

NB-Typ	Schlupf/Ueberschuss	Kuenstliche Var.
\leq	$+s_i$	nicht noetig
\geq	$-s_i$	$+a_i$ noetig
$=$	$-$	$+a_i$ noetig

Umgewandelt: maximize $z' = -2x_1 - 3x_2 - Ma_1$ mit $x_1 + x_2 - s_1 + a_1 = 4$, $2x_1 + x_2 + s_2 = 10$.

Start: $a_1 = 4$, $s_2 = 10$, $z' = -4M$.

Wichtig: Vor Start z-Zeile korrigieren (Gauss-Schritt), damit a_1 -Spalte Eintrag 0 hat.

Die kuenstliche Variable ist ein Hilfsgeruest – am Ende muss $a_j = 0$ gelten.

Korrigiertes Ausgangstableau:

Basis	x_1	x_2	s_1	s_2	a_1	RHS
a_1	1	1	-1	0	1	4
s_2	2	1	0	1	0	10
z'	$-2 + M$	$-3 + M$	M	0	0	$-4M$

Pivot: x_2 tritt ein, a_1 tritt aus. Nach weiteren Pivots:

Optimum: $x_1^* = 0$, $x_2^* = 10$, $z_{\min} = 6$, $a_1 = 0$ ✓.

Die Big-M-Methode fügt genau so viele kuenstliche Variablen ein wie \geq - und $=$ -NB vorhanden sind.

Vorteile:

- Einfach zu verstehen
- Nur eine Phase noetig

Nachteile:

- M zu gross \rightarrow numerische Instabilitaet
- M zu klein \rightarrow falsche Loesung

Alternative: Zweiphasen-Methode

- Phase I: minimize $\sum a_i$ (Hilfsproblem)
- Phase II: Originalproblem mit zulaessiger Basis
- Kein grosses $M \Rightarrow$ numerisch stabiler

Moderne Solver nutzen weder Big-M noch Zweiphasen – stattdessen Crash-Heuristiken und Preprocessing.

Ziel

Finde eine zulaessige Basisloesung fuer das urspruengliche LP, indem ein **Hilfsproblem** geloest wird.

Hilfsproblem:

$$\begin{aligned} \text{minimize} \quad & w = a_1 + a_2 + \dots + a_k \\ \text{s.t.} \quad & \text{Originale NB mit kuenstlichen Var.} \\ & \text{alle Variablen} \geq 0 \end{aligned}$$

Unser Beispiel:

$$\begin{aligned} \text{minimize} \quad & w = a_1 \\ \text{s.t.} \quad & x_1 + x_2 - s_1 + a_1 = 4 \\ & 2x_1 + x_2 + s_2 = 10 \end{aligned}$$

Start: $a_1 = 4, s_2 = 10, w = 4$.

Phase I beantwortet: "Ist das LP zulaessig?" – eine fundamentale Frage, die Big-M nur implizit beantwortet.

Moegliche Ergebnisse nach Phase I:

1. $w^* = 0$: Alle $a_i = 0$.
⇒ Zulaessige Basis gefunden! Weiter mit Phase II.
2. $w^* > 0$: Mindestens ein $a_i > 0$.
⇒ Das LP ist **unzulaessig** ($\mathcal{F} = \emptyset$). **Stopp**.

Vorteil gegenueber Big-M:

- Kein Parameter M noetig
- Numerisch stabil
- Unzulaessigkeit wird klar erkannt

Voraussetzung

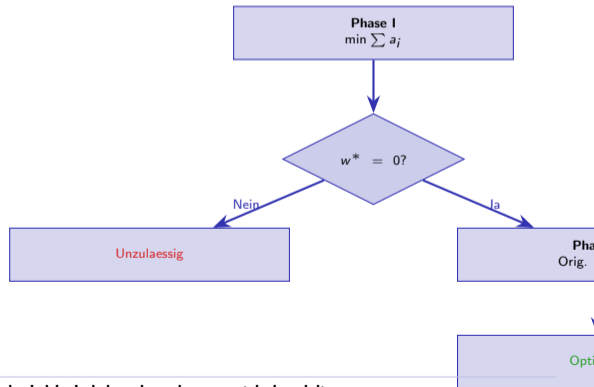
Phase I hat $w^* = 0$ ergeben \Rightarrow alle kuenstlichen Variablen sind 0.

Vorgehen:

1. Entferne alle kuenstlichen Variablen (a_i) aus dem Tableau.
2. Ersetze die Zielfunktionszeile durch die **originale** Zielfunktion.
3. Korrigiere die z-Zeile: Eliminiere Basisvariablen-Eintraege mittels Gauss.
4. Fuehre den normalen Simplex durch.

Ergebnis: Die Optimalloesung des urspruenglichen LP.

Ablauf als Flussdiagramm:



Die Zweiphasen-Methode ist die theoretisch sauberere Alternative zu Big-M und wird in Lehrbuechern bevorzugt behandelt.

Vergleich: Big-M vs Zweiphasen-Methode

Kriterium	Big-M	Zweiphasen
Einfachheit	✓ einfacher	komplexer
Numerische Stabilität	× problematisch	✓ stabil
Parameterwahl	M muss gewählt werden	kein Parameter
Erkennung Unzulässigkeit	indirekt ($a_i > 0$)	direkt ($w^* > 0$)
Anzahl Phasen	1	2
Praxis-Einsatz	selten	theoretisch bevorzugt
Moderne Solver	keines von beiden	Crash-Heuristiken

Empfehlung

Für Lehre und Handrechnung: **Big-M** (einfacher).

Für Theorie und Beweise: **Zweiphasen** (sauberer).

Für Praxis: **Weder noch** – Solver verwenden fortgeschrittene Techniken.

Beide Methoden lösen dasselbe Problem: Wie finde ich eine zulässige Startbasis, wenn der Ursprung nicht zulässig ist?

Primales LP:

$$\begin{aligned} & \underset{x}{\text{maximize}} && \mathbf{c}^T \mathbf{x} \\ & \text{s.t.} && \mathbf{Ax} \leq \mathbf{b}, \mathbf{x} \geq \mathbf{0} \end{aligned}$$

\iff

Duales LP:

$$\begin{aligned} & \underset{y}{\text{minimize}} && \mathbf{b}^T \mathbf{y} \\ & \text{s.t.} && \mathbf{A}^T \mathbf{y} \geq \mathbf{c}, \mathbf{y} \geq \mathbf{0} \end{aligned}$$

Merkregel: $\mathbf{A} \rightarrow \mathbf{A}^T$, $\mathbf{c} \leftrightarrow \mathbf{b}$, $\max \leftrightarrow \min$, $\leq \leftrightarrow \geq$.

Das Dual des Duals ist wieder das Primal – symmetrische Beziehung.

Satz

Fuer zulaessige \mathbf{x} (primal) und \mathbf{y} (dual): $\mathbf{c}^T \mathbf{x} \leq \mathbf{b}^T \mathbf{y}$.

Beweis: $\mathbf{c}^T \mathbf{x} \leq (\mathbf{A}^T \mathbf{y})^T \mathbf{x} = \mathbf{y}^T \mathbf{A} \mathbf{x} \leq \mathbf{y}^T \mathbf{b}$.

Konsequenz: Duale Loesung = obere Schranke fuer Primal.

Falls $\mathbf{c}^T \mathbf{x}^* = \mathbf{b}^T \mathbf{y}^*$, sind beide optimal – das ist der starke Dualitaetssatz.

Definition

Schattenpreis $y_i^* = \frac{\partial z^*}{\partial b_i}$: Grenzwert einer zusätzlichen Einheit der Ressource i .

Bäckerei:

- $y_1^* = \frac{9}{4}$ (Mehl): +1 kg \rightarrow +2.25 Euro
- $y_2^* = \frac{1}{4}$ (Arbeit): +1 Std. \rightarrow +0.25 Euro

Mehl ist 9-mal wertvoller!

Ablezen: z -Zeile unter Schlupfvariablen.

Nichtaktive NB: $s_i > 0 \Rightarrow y_i^* = 0$ (Ressource nicht knapp).

Schattenpreise gelten nur im Gültigkeitsbereich (RHS-Ranging).

Primal: $\max 3x_1 + 5x_2$, NB wie bekannt.
 $x^* = (2, 3)$, $z^* = 21$.

Dual: $\min 8y_1 + 12y_2$, $y_1 + 3y_2 \geq 3$, $2y_1 + 2y_2 \geq 5$.
 $y^* = (\frac{9}{4}, \frac{1}{4})$, $w^* = 21$.

Starke Dualitaet

$$z^* = 21 = w^* \checkmark$$

Deutung: Primal = Produzent (max Gewinn). Dual = Ressourcenbewerter (min Ressourcenwert bei fairem Preis).

Die duale Loesung faellt beim Simplex gratis ab – aus dem Optimaltableau ablesbar.

Satz

Wenn Primal optimal \Rightarrow Dual optimal, und $\mathbf{c}^T \mathbf{x}^* = \mathbf{b}^T \mathbf{y}^*$.

Alle Faelle:

	Dual optimal	Dual unzuessaessig
Primal optimal	$z^* = w^*$	unmoeglich
Primal unzuessaessig	unmoeglich	moeglich
Primal unbeschraenkt	unmoeglich	ja

Der starke Dualitaetssatz verbindet Primal und Dual vollstaendig.

Satz (Complementary Slackness)

$y_i^*(b_i - \mathbf{a}_i^T \mathbf{x}^*) = 0$ und $x_j^*(\mathbf{a}_j^T \mathbf{y}^* - c_j) = 0$ fuer alle i, j .

Bedeutung: "Duale Variable positiv \Leftrightarrow Primale NB aktiv."

Baeckerei: $y_1^* = \frac{9}{4} > 0$ und $s_1 = 0$ ✓; $y_2^* = \frac{1}{4} > 0$ und $s_2 = 0$ ✓.

Komplementärer Schlupf erlaubt, die duale Loesung aus der primalen zu berechnen.

Primal (Produzent):

- Entscheidet Produktionsmengen x
- Maximiert Gewinn
- Beschraenkt durch Ressourcen

Dual (Ressourcenkaeufer):

- Bietet Preise y fuer Ressourcen
- Minimiert Kaufpreis
- Faire Preise: $\mathbf{A}^T \mathbf{y} \geq \mathbf{c}$

Gleichgewicht: Max-Gewinn = Min-Ressourcenwert. Mehlpreis: 2.25 Euro/kg, Arbeitspreis: 0.25 Euro/Std.

Duale Interpretation gibt Managern Aufschluss, welche Ressourcen wertvoll sind.

Satz

Hat das LP $\max \mathbf{c}^T \mathbf{x}$ mit $\mathbf{x} \in \mathcal{F}$ (konvexes Polytop) eine Lösung, so liegt eine am Eckpunkt.

Beweis:

1. Sei $F^* = \{\mathbf{x} \in \mathcal{F} : \mathbf{c}^T \mathbf{x} = z^*\}$ die **Optimalflaeche**. F^* ist ein konvexes Polytop (Schnitt von \mathcal{F} mit Hyperebene).
2. Jedes konvexe Polytop besitzt mindestens einen Eckpunkt (Satz ueber Polytope).
3. Sei \mathbf{v} ein Eckpunkt von F^* . Dann ist \mathbf{v} auch Eckpunkt von \mathcal{F} :
 - Annahme: $\mathbf{v} = \lambda \mathbf{u} + (1 - \lambda) \mathbf{w}$ mit $\mathbf{u}, \mathbf{w} \in \mathcal{F}$, $0 < \lambda < 1$.
 - $z^* = \mathbf{c}^T \mathbf{v} = \lambda \mathbf{c}^T \mathbf{u} + (1 - \lambda) \mathbf{c}^T \mathbf{w} \leq \lambda z^* + (1 - \lambda) z^* = z^*$.
 - Gleichheit erzwingt $\mathbf{c}^T \mathbf{u} = \mathbf{c}^T \mathbf{w} = z^*$, also $\mathbf{u}, \mathbf{w} \in F^*$.
 - Da \mathbf{v} Eckpunkt von F^* : $\mathbf{u} = \mathbf{w} = \mathbf{v}$, also \mathbf{v} Eckpunkt von \mathcal{F} .

□

Der Beweis nutzt Konvexität und die Polytop-Eigenschaft – er generalisiert auf beliebige LP in jeder Dimension.

Satz

Hat das primale LP eine optimale Loesung \mathbf{x}^* , so auch das duale, und $\mathbf{c}^T \mathbf{x}^* = \mathbf{b}^T \mathbf{y}^*$.

Beweisskizze (ueber Simplex):

1. Der Simplex-Algorithmus terminiert mit optimalem Tableau.
2. Im Optimaltableau: $\bar{c}_j \geq 0$ fuer alle j .
3. Setze $y_i^* = \bar{c}_{s_i}$ (Eintraege unter Schlupfvariablen in der z-Zeile).
4. **Duale Zulaessigkeit:** $\mathbf{A}^T \mathbf{y}^* \geq \mathbf{c}$ folgt aus $\bar{c}_j = c_j - \mathbf{c}_B^T \mathbf{B}^{-1} \mathbf{a}_j \geq 0$.
5. **Komplementaerer Schlupf:** Basisvariablen haben $\bar{c}_j = 0$, Nichtbasis hat $x_j = 0$.
6. **Starke Dualitaet:** $\mathbf{b}^T \mathbf{y}^* = \mathbf{c}_B^T \mathbf{B}^{-1} \mathbf{b} = \mathbf{c}_B^T \mathbf{x}_B^* = \mathbf{c}^T \mathbf{x}^*$. □

Der Beweis zeigt: Der Simplex liefert die duale Loesung gratis mit – keine separate Berechnung noetig.

Satz

Im nichtdegenerierten Fall terminiert der Simplex-Algorithmus nach endlich vielen Schritten.

Beweis:

1. Es gibt höchstens $\binom{n+m}{m}$ Basen.
2. Jeder Pivotschritt ändert die Basis.
3. Im nichtdegenerierten Fall: z steigt **strikt** in jedem Schritt.
4. Keine Basis wird zweimal besucht.
5. \Rightarrow Terminierung nach $\leq \binom{n+m}{m}$ Schritten. □

Degenerierter Fall:

- z kann gleich bleiben (Seitensprung).
- Theoretisch: Zyklen möglich.
- **Bland-Regel:** Garantiert Terminierung auch bei Degenerierung.

Praxis:

Zyklen sind extrem selten. Moderne Solver implementieren Anti-Cycling als Sicherheitsnetz.

Die Endlichkeit ist eine wichtige theoretische Eigenschaft – in der Praxis ist die Anzahl der Pivots typisch $O(m)$.

Simplex:

- **Worst Case:** Exponentiell.
Klee-Minty (1972): $2^n - 1$ Pivots bei n Dimensionen.
- **Average Case:** Polynomial.
Borgwardt (1982): $O(n^2 m^{1/(n-1)})$ im Mittel.
- **Smoothed Analysis:** Polynomial.
Spielman & Teng (2004): $O(\text{poly}(n, m, 1/\sigma))$.
- **Praxis:** Typisch $O(m)$ bis $O(m^2)$ Pivots.

Polynomiale Alternativen:

- **Ellipsoid (Khachiyan, 1979):**
 $O(n^4 L)$, aber praktisch langsam.
- **Interior Point (Karmarkar, 1984):**
 $O(n^{3.5} L)$, praktisch schnell.
- $L =$ Eingabelaenge in Bits.

Fazit:

- LP \in P (polynomiale Klasse)
- Simplex: praktisch am schnellsten fuer die meisten Instanzen
- Interior Point: konkurrenzfaehig bei sehr grossen LPs

Die Frage, ob ein polynomialer Simplex-Variante existiert, ist offen – eines der grossen offenen Probleme der Optimierung.

Fabrik mit 3 Maschinen, 2 Produkte.

	Maschine A	Maschine B	Maschine C
P_1	1	3	1
P_2	2	2	0
Kapazitaet	8	18	5
Gewinn	P_1 : 4 Euro, P_2 : 5 Euro		

LP: $\max 4x_1 + 5x_2$ mit $x_1 + 2x_2 \leq 8$, $3x_1 + 2x_2 \leq 18$, $x_1 \leq 5$, $x_1, x_2 \geq 0$.

Produktionsplanung skaliert auf Hunderte von Produkten und Maschinen.

m Lager, n Kunden: $\min \sum c_{ij}x_{ij}$ mit Angebots- und Nachfragebeschränkungen.

	K_1	K_2	K_3	Angebot
L_1	2	3	1	20
L_2	5	4	8	30
Nachfrage	10	15	15	

Spezialstruktur: Transportalgorithmus effizienter als Standard-Simplex. Ganzzahlige Lösung bei ganzzahligem Input.

Transportproblem: Spezialfall des LP mit besonders effizienter Netzwerkstruktur.

$$\max \sum r_j x_j \text{ mit } \sum x_j = 1, x_j \leq u_j, x_j \geq 0.$$

Asset	r_j	u_j
Aktien	8%	0.40
Anleihen	3%	0.50
Immobilien	5%	0.30

Hinweis: Markowitz = QP (Varianzminimierung). LP-Modelle nutzen lineare Risikomasse (CVaR, MAD).

LP-Portfoliomodelle sind in der Praxis weit verbreitet, insbesondere mit CVaR als Risikomass.

Code

```
from scipy.optimize import linprog

c = [-3, -5] # negativ fuer max
A_ub = [[1, 2], [3, 2]]
b_ub = [8, 12]
bounds = [(0, None), (0, None)]

res = linprog(c, A_ub=A_ub, b_ub=b_ub,
              bounds=bounds, method='highs')

print(f"x* = {res.x}")
print(f"z* = {-res.fun:.1f}")
print(f"Schlupf: {res.slack}")
```

Ausgabe

```
x* = [2. 3.]
z* = 21.0
Schlupf: [0. 0.]
```

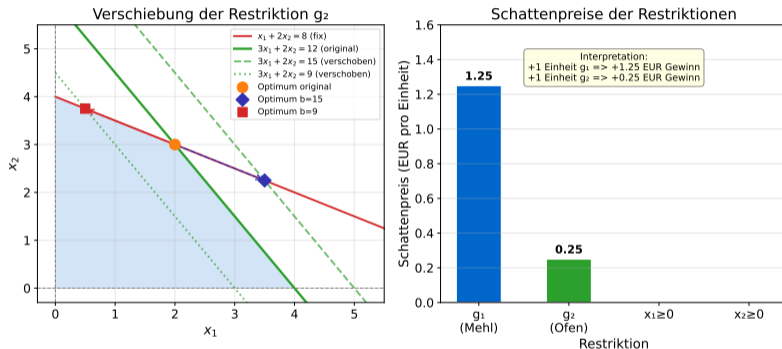
Hinweise:

- linprog minimiert
- method='highs': moderner Solver
- Fuer grosse LPs: PuLP, Gurobi, CPLEX

scipy.optimize.linprog ist ideal fuer Prototyping – fuer Produktion spezialisierte Solver nutzen.

Schattenpreise im Detail

$y_i^* = \frac{\Delta z^*}{\Delta b_i}$ (lokal). Bäckerei: $y_1^* = 2.25$ (Mehl), $y_2^* = 0.25$ (Arbeit).



Im Gültigkeitsbereich ist z^* linear in b_i mit Steigung y_i^* .

Schattenpreise = Lagrange-Multiplikatoren der Nebenbedingungen.

Frage: In welchem Bereich darf b_i variieren, sodass die Basis erhalten bleibt?

Methode: $\mathbf{B}^{-1}\mathbf{b} \geq \mathbf{0}$ muss gelten. Fuer $b_1 = 8 + \Delta$:

- $x_2 = 3 + \frac{3}{4}\Delta \geq 0 \Rightarrow \Delta \geq -4$
- $x_1 = 2 - \frac{1}{2}\Delta \geq 0 \Rightarrow \Delta \leq 4$

$\Rightarrow b_1 \in [4, 12]$: Schattenpreis $y_1^* = 2.25$ gilt uneingeschraenkt.

RHS-Ranging ist Standardoutput professioneller Solver.

Frage: Wie weit darf c_j variieren, sodass die Basisloesung optimal bleibt?

Methode: Alle $\bar{c}_j \geq 0$ muessen gelten.

Brotgewinn $c_1 = 3$: Darf bis 2.50 sinken, ohne dass sich die Produktionsentscheidung aendert.

Geometrisch: c_j dreht die Isoprofit-Linie. Solange dieselbe Ecke zuletzt beruehrt wird, bleibt die Basis gleich.

Koeffizientenanalyse zeigt Robustheit gegenueber Gewinn-/Kostenschwankungen.

Variable	Wert	Red. Kosten	c_j min	c_j max
x_1	2	0	2.50	∞
x_2	3	0	0	6.00

NB	Schlupf	Schatten	b_i min	b_i max
Mehl	0	2.25	4	12
Arbeit	0	0.25	8	24

Sensitivitätsanalyse liefert dem Management direkt umsetzbare Handlungsempfehlungen.

Definition

Ein **ganzzahliges LP** (Integer Program, IP) verlangt, dass einige oder alle Variablen ganzzahlige Werte annehmen:

$$\mathbf{x} \in \mathbb{Z}^n \quad (\text{oder gemischt: einige } x_j \in \mathbb{Z}, \text{ andere } x_j \in \mathbb{R})$$

Anwendungen:

- Standortplanung (ja/nein-Entscheidungen)
- Scheduling (ganzzahlige Zuordnungen)
- Investitionsentscheidungen (binaer)
- Stueckzahl-Planung

Typen:

- **IP:** Alle Variablen ganzzahlig
- **MIP:** Gemischt-ganzzahlig
- **BIP:** Binaer ($x_j \in \{0, 1\}$)

Schwierigkeit:

- IP ist **NP-schwer** (im Allgemeinen)
- Einfaches Runden der LP-Relaxation liefert oft keine zulaessige oder optimale Loesung
- Zulaessiger Bereich ist **nicht konvex**

Beispiel:

LP-Relaxation: $x^* = 2.7$

Runden auf 3: moeglicherweise unzulaessig!

Runden auf 2: moeglicherweise nicht optimal!

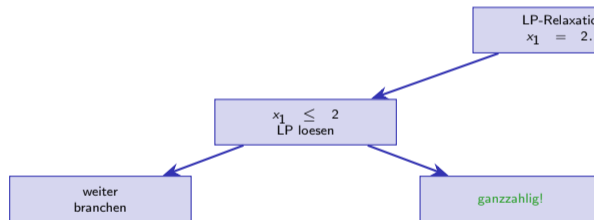
LP ist das Fundament fuer IP: Jeder IP-Solver loest intern viele LPs (als Relaxationen).

Algorithmus

Branch-and-Bound loest ein IP, indem es systematisch LP-Relaxationen loest und den Suchraum einschraenkt.

Vorgehen:

1. Loese die **LP-Relaxation** (ohne Ganzzahligkeit).
2. Ist die Loesung ganzzahlig? \Rightarrow Fertig!
3. Sonst: **Branch** – waehle eine nicht-ganzzahlige Variable $x_j = f$ und erzeuge zwei Teilprobleme:
 - $x_j \leq \lfloor f \rfloor$
 - $x_j \geq \lceil f \rceil$
4. **Bound** – LP-Relaxation liefert obere Schranke. Wenn schlechter als beste bekannte ganzzahlige Loesung: **prunen** (abschneiden).



Praxis: Solver wie Gurobi loesen MIPs mit Millionen Variablen durch B&B + Schnittebenen + Heuristiken.

Branch-and-Bound nutzt LP-Relaxationen als Schranken – je besser die Relaxation, desto schneller die Loesung.

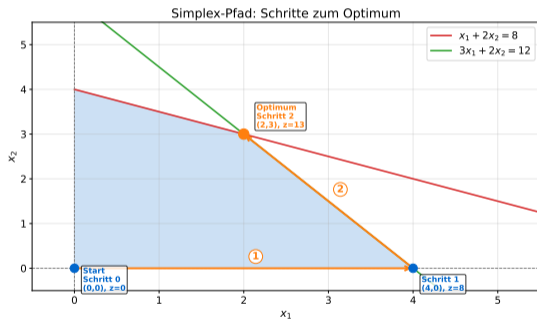
Grundidee

Statt ueber Kanten (Simplex) bewegt sich die Innere-Punkte-Methode **durch das Innere** des Polyeders zum Optimum.

Prinzip:

1. Starte im Inneren von \mathcal{F} .
2. Verwende eine **Barrierefunktion**:
$$B(\mathbf{x}) = \mathbf{c}^T \mathbf{x} - \mu \sum_i \ln(b_i - \mathbf{a}_i^T \mathbf{x})$$
3. Reduziere $\mu \rightarrow 0$ schrittweise.
4. Im Limes $\mu \rightarrow 0$: Optimum am Rand.

Komplexitaet: $O(n^{3.5}L)$ – polynomial!



Varianten:

- Primal-Dual Interior Point
- Predictor-Corrector (Mehrotra)

Vergleich: Simplex vs Interior Point

Kriterium	Simplex	Interior Point
Pfad	Kanten (Eckpunkte)	Inneres (zentraler Pfad)
Worst Case	exponentiell	polynomial
Praxis (klein)	sehr schnell	gut
Praxis (gross)	schnell	sehr schnell
Warmstart	effizient	schwierig
Basisloesung	direkt	Nachbearbeitung
Sensitivitaet	direkt	Nachbearbeitung
Degenerierung	problematisch	kein Problem

Praxis: Moderne Solver (Gurobi, CPLEX, HiGHS) bieten **beide** und waehlen automatisch. Fuer Sensitivity und Warmstart: Simplex bevorzugt.

Die Kombination beider Methoden ist der Stand der Technik in kommerziellen LP-Solvern.

Vollständiger Code

```
from scipy.optimize import linprog

c = [-3, -5]
A_ub = [[1, 2], [3, 2]]
b_ub = [8, 12]
bounds = [(0, None), (0, None)]

res = linprog(c, A_ub=A_ub, b_ub=b_ub,
              bounds=bounds, method='highs')

print("Status:", res.message)
print("x* =", res.x)
print("z* =", -res.fun)
print("Schlupf:", res.slack)
```

Rueckgabe:

- `res.x`: x^*
- `res.fun`: Optimalwert (min!)
- `res.slack`: Schlupfwerte
- `res.status`: 0 = Erfolg

Status-Codes: 0=OK, 2=unzulaessig,
3=unbeschraenkt.

HiGHS in scipy kombiniert Simplex und Interior Point – ab scipy 1.9 der Standard.

Gleichungs-NB und Schranken:

```
A_eq = [[1, 1]]  
b_eq = [5]  
bounds = [(0, 4), (0, 3)]
```

```
res = linprog(c,  
             A_ub=A_ub, b_ub=b_ub,  
             A_eq=A_eq, b_eq=b_eq,  
             bounds=bounds)
```

Fuer groessere Probleme:

- **PuLP**: Modellierungssprache
- **Pyomo**: Umfangreiches Framework
- **gurobipy**: Gurobi Python API
- **CVXPY**: Convex Optimization

Pruefen Sie immer den Status-Code – “Optimum gefunden” ist keine Selbstverstaendlichkeit.

Problem

In einem gerichteten Graphen soll ein maximaler Fluss von einer Quelle s zu einer Senke t geschickt werden, unter Kapazitätsbeschränkungen u_{ij} für jede Kante (i, j) .

LP-Formulierung:

$$\begin{aligned} & \text{maximize} && \sum_j f_{sj} \\ & \text{s.t.} && 0 \leq f_{ij} \leq u_{ij} \\ & && \sum_j f_{ij} = \sum_j f_{ji} \quad \forall i \neq s, t \end{aligned}$$

(Flusserhaltung an jedem Knoten)

Spezialfälle:

- **Max-Flow / Min-Cut:** Dualität liefert minimalen Schnitt
- **Kürzeste Wege:** LP-Formulierung möglich
- **Zuordnungsproblem:** Bipartites Matching als LP

Alle diese Probleme sind Spezialfälle von LP mit ganzzahligen Lösungen (Totalunimodularität).

Netzwerkflussprobleme sind fundamental in der Graphentheorie und lösbar in stark polynomialer Zeit.

Nullsummenspiel

Zwei Spieler mit Auszahlungsmatrix $\mathbf{A} \in \mathbb{R}^{m \times n}$. Spieler 1 wählt Zeile i , Spieler 2 wählt Spalte j . Auszahlung: a_{ij} an Spieler 1.

Spieler 1 (Maximierer):

$$\begin{aligned} &\text{maximize } v \\ &\text{s.t. } \sum_i a_{ij} x_i \geq v \quad \forall j \\ &\quad \sum_i x_i = 1, \quad x_i \geq 0 \end{aligned}$$

x_i : Wahrscheinlichkeit fuer Zeile i .

Spieler 2 (Minimierer):

$$\begin{aligned} &\text{minimize } w \\ &\text{s.t. } \sum_j a_{ij} y_j \leq w \quad \forall i \\ &\quad \sum_j y_j = 1, \quad y_j \geq 0 \end{aligned}$$

y_j : Wahrscheinlichkeit fuer Spalte j .

Minimax-Satz (von Neumann): $v^* = w^*$ – Starke Dualitaet!

Die LP-Formulierung der Spieltheorie war einer der Ausgangspunkte fuer die Dualitaetstheorie (von Neumann, 1947).

Problem

Minimiere die **Summe der absoluten Abweichungen** statt der quadratischen (wie bei OLS):

$$\underset{\beta}{\text{minimize}} \quad \sum_{i=1}^n |y_i - \mathbf{x}_i^T \beta|$$

LP-Umformulierung:

Fuehre Hilfsvariablen $u_i, v_i \geq 0$ ein mit $y_i - \mathbf{x}_i^T \beta = u_i - v_i$:

$$\begin{aligned} \underset{}{\text{minimize}} \quad & \sum_{i=1}^n (u_i + v_i) \\ \text{s.t.} \quad & \mathbf{x}_i^T \beta + u_i - v_i = y_i \\ & u_i, v_i \geq 0 \end{aligned}$$

(β frei, ggf. Substitution)

Vorteile gegenueber OLS:

- **Robuster:** Weniger empfindlich gegenueber Ausreissern
- L_1 -Norm bestraft grosse Abweichungen weniger stark als L_2
- Loesungen sind **duenner** (Sparse): Einige Residuen exakt = 0

Verbindung zu ML:

LASSO-Regression (L_1 -Regularisierung) nutzt dasselbe Prinzip fuer Feature-Selektion.

L_1 -Regression verbindet klassische Statistik mit linearer Optimierung – und ist Grundlage fuer moderne Sparse-Methoden.

Aufgabe 1: LP formulieren

Modellierung:

Eine Schreinerei stellt Tische (x_1) und Stuehle (x_2) her.

- Gewinn: Tisch 80 Euro, Stuhl 50 Euro.
- Holz: Tisch 6 m, Stuhl 2 m. Verfuegbar: 36 m.
- Arbeitszeit: Tisch 4 Std., Stuhl 3 Std. Verfuegbar: 24 Std.
- Nachfrage Stuehle: hoechstens 8.

Aufgaben:

1. Formulieren Sie das LP in Standardform.
2. Bestimmen Sie \mathbf{c} , \mathbf{A} , \mathbf{b} .
3. Wie viele Entscheidungsvariablen und Nebenbedingungen hat das Problem?

Loesung (Kurzform)

maximize $80x_1 + 50x_2$ mit $6x_1 + 2x_2 \leq 36$, $4x_1 + 3x_2 \leq 24$, $x_2 \leq 8$, $x_1, x_2 \geq 0$.

$\mathbf{c} = (80, 50)^T$, $\mathbf{A} = \begin{pmatrix} 6 & 2 \\ 4 & 3 \\ 0 & 1 \end{pmatrix}$, $\mathbf{b} = (36, 24, 8)^T$. 2 Variablen, 3 NB + 2 Nichtneg.

Modellierung ist oft der schwierigste Schritt – uebersetzen Sie die Sachbeschreibung sorgfaeltig in Mathematik.

Aufgabe 2: Graphisch loesen

Loesen Sie graphisch:

$$\begin{aligned} \text{maximize} \quad & z = 2x_1 + 3x_2 \\ \text{s.t.} \quad & x_1 + x_2 \leq 6 \\ & 2x_1 + x_2 \leq 10 \\ & x_1 \leq 4 \\ & x_1, x_2 \geq 0 \end{aligned}$$

Schritte:

1. Zeichnen Sie den zulaessigen Bereich (alle NB als Geraden, Schnittmenge).
2. Bestimmen Sie alle Eckpunkte: $(0, 0)$, $(4, 0)$, $(4, 2)$, $(?, ?)$, $(0, 6)$.
3. Werten Sie z an jedem Eckpunkt aus.
4. Zeichnen Sie Isoprofit-Linien $z = 6, 12, 18$.
5. Lesen Sie das Optimum ab und verifizieren Sie die Zulaessigkeit.

Hinweis

Eckpunkt C: Schnitt von $x_1 + x_2 = 6$ und $2x_1 + x_2 = 10 \Rightarrow x_1 = 4, x_2 = 2$. Aber auch $x_1 \leq 4$ pruefen!

Graphische Loesung trainiert die geometrische Intuition – wichtig fuer das Verstaendnis des Simplex.

Aufgabe 3: Simplex-Tableau durchrechnen

Loesen Sie mit dem Simplex-Verfahren:

$$\begin{aligned} \text{maximize} \quad & z = 5x_1 + 4x_2 \\ \text{s.t.} \quad & x_1 + x_2 \leq 5 \\ & 2x_1 + x_2 \leq 8 \\ & x_1, x_2 \geq 0 \end{aligned}$$

Schritte:

1. Fuehren Sie Schlupfvariablen ein und stellen Sie das Ausgangstableau auf.
2. Bestimmen Sie Pivotspalte und Pivotzeile.
3. Fuehren Sie den Pivotschritt durch.
4. Pruefen Sie die Optimalitaet.
5. Falls nicht optimal: Wiederholen.
6. Lesen Sie die Optimalloesung ab und verifizieren Sie.

Ausgangstableau:

Basis	x_1	x_2	s_1	s_2	RHS
s_1	1	1	1	0	5
s_2	2	1	0	1	8
z	-5	-4	0	0	0

Erwartetes Ergebnis: $x_1^* = 3$, $x_2^* = 2$, $z^* = 23$. Ueberpruefen Sie!

Aufgabe 4: Dualitaet und Interpretation

Gegeben: Das primale LP aus Aufgabe 3 mit Optimaltableau.

Aufgaben:

1. Stellen Sie das **duale LP** auf (Variablen, Zielfunktion, NB).
2. Lesen Sie die **duale Optimalloesung** aus dem primalen Optimaltableau ab.
3. Verifizieren Sie den **starken Dualitaetssatz** ($z^* = w^*$).
4. Pruefen Sie den **komplementaeren Schlupf** fuer alle Variablen.
5. Interpretieren Sie die **Schattenpreise** oekonomisch:
 - Welche Nebenbedingung ist wertvoller?
 - Was passiert, wenn die rechte Seite um 1 erhoehrt wird?
6. **Bonus:** Bestimmen Sie den RHS-Ranging-Bereich fuer beide Nebenbedingungen.

Hinweis zum Dual

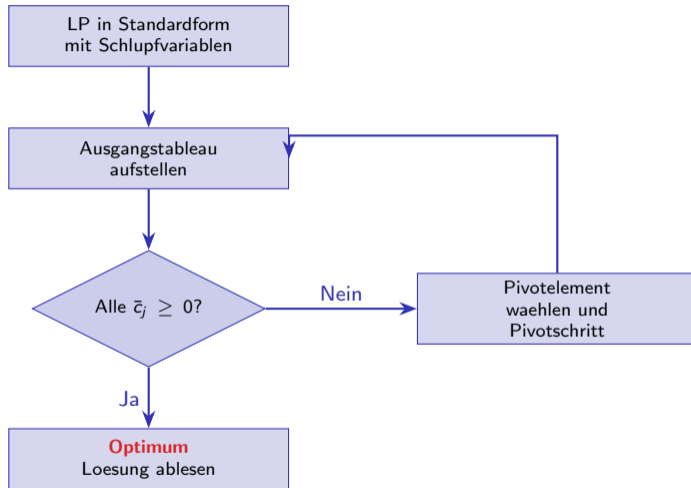
Primal: $\max 5x_1 + 4x_2, \mathbf{Ax} \leq \mathbf{b}$.

Dual: $\min 5y_1 + 8y_2, y_1 + 2y_2 \geq 5, y_1 + y_2 \geq 4, y_1, y_2 \geq 0$.

Die Schattenpreise stehen in der z-Zeile unter s_1 und s_2 im Optimaltableau.

Diese Aufgabe verbindet Simplex, Dualitaet und Sensitivitaetsanalyse zu einem Gesamtbild.

Simplex-Algorithmus: Flussdiagramm



Fuer polynomiale Worst-Case-Garantie: Innere-Punkte-Methoden (Karmarkar, 1984).

LP-Standardform: maximize $\mathbf{c}^T \mathbf{x}$, $\mathbf{Ax} \leq \mathbf{b}$, $\mathbf{x} \geq \mathbf{0}$

Schlupf: $\mathbf{Ax} + \mathbf{s} = \mathbf{b}$

Simplex: Pivotspalte = $\min \bar{c}_j < 0$, Pivotzeile = $\min \frac{b_i}{a_{ij}}$

Optimal: $\bar{c}_j \geq 0 \forall j$

Big-M: a_i mit Strafe M in Zielfunktion

Dual: min $\mathbf{b}^T \mathbf{y}$, $\mathbf{A}^T \mathbf{y} \geq \mathbf{c}$

Schwach: $\mathbf{c}^T \mathbf{x} \leq \mathbf{b}^T \mathbf{y}$

Stark: $z^* = w^*$

Kompl. Schlupf: $y_i^* s_i^* = 0$

Schattenpreis: $y_i^* = \partial z^* / \partial b_i$

Interior Point: Barriere $-\mu \sum \ln s_i$

Beherrschen Sie diese Formeln fuer Klausur und Praxis.

Einfuehrung:

- **Bertsimas & Tsitsiklis:** *Introduction to Linear Optimization*, Athena Scientific, 1997. (Standardwerk)
- **Chvatal:** *Linear Programming*, W. H. Freeman, 1983. (Klassiker, sehr klar)
- **Vanderbei:** *Linear Programming*, 5th ed., Springer, 2020. (Modern, mit Algorithmen)

Fortgeschritten:

- **Dantzig & Thapa:** *Linear Programming*, Springer, 1997/2003. (Vom Erfinder des Simplex)
- **Schrijver:** *Theory of Linear and Integer Programming*, Wiley, 1998. (Theorie)

Ganzzahlige Optimierung:

- **Wolsey:** *Integer Programming*, 2nd ed., Wiley, 2020.
- **Nemhauser & Wolsey:** *Integer and Combinatorial Optimization*, Wiley, 1988.

Software-Dokumentation:

- `scipy.optimize.linprog`: scipy.org
- PuLP: github.com/coin-or/pulp
- Gurobi: gurobi.com
- HiGHS: highs.dev

Online-Kurse:

- MIT OCW 15.053 (Optimization Methods)
- Stanford EE364a (Convex Optimization)

Lineare Optimierung ist das Fundament fuer Machine Learning, Operations Research und Supply-Chain-Management.