

Lineare Optimierung

Was steckt dahinter? – Einfuehrung

BSc Analysis

2025

Kennen Sie dieses Problem?

**Sie haben 100 Euro fuer Lebensmittel.
Sie wollen so SATT wie moeglich werden.**

Brot

Billig, macht satt
aber langweilig

Obst

Teurer, gesund
aber weniger saettigend

Fleisch

Am teuersten
aber sehr nahrhaft

Was kaufen Sie? Wie teilen Sie Ihr Budget auf?

DAS ist lineare Optimierung: Das Beste aus begrenzten Mitteln machen.

Jedes Unternehmen loest taeglich genau solche Probleme – nur mit Millionen-Budgets

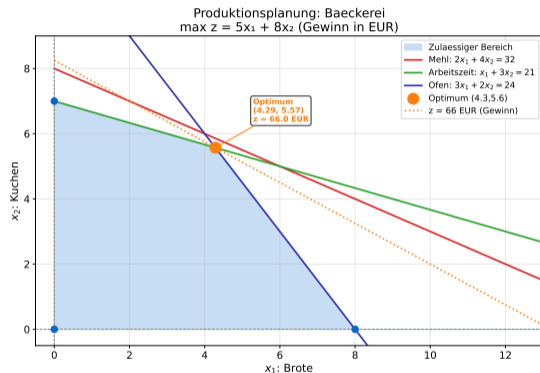
Die Bäckerei – Ein konkretes Beispiel

Die Situation

- Eine Bäckerei stellt **Brote** und **Kuchen** her
- **Brot**: braucht 2 kg Mehl, 1 Stunde Backzeit
- **Kuchen**: braucht 1 kg Mehl, 2 Stunden Backzeit
- Verfügbar: **8 kg Mehl** und **8 Stunden**

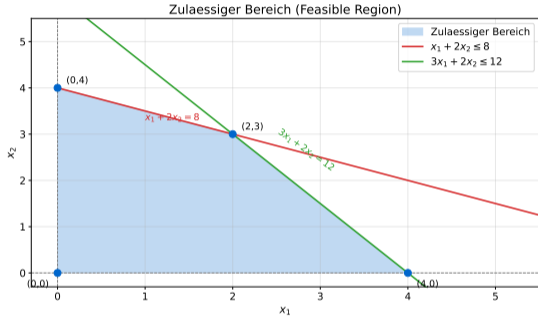
Das Ziel

- Brot bringt **3 EUR** Gewinn pro Stueck
- Kuchen bringt **5 EUR** Gewinn pro Stueck
- **Frage**: Wie viele Brote und Kuchen backen fuer **maximalen Gewinn**?



Das klingt einfach – aber mit 100 Produkten und 50 Maschinen wird es spannend

Zeichnen wir es!



Was sehen wir hier?

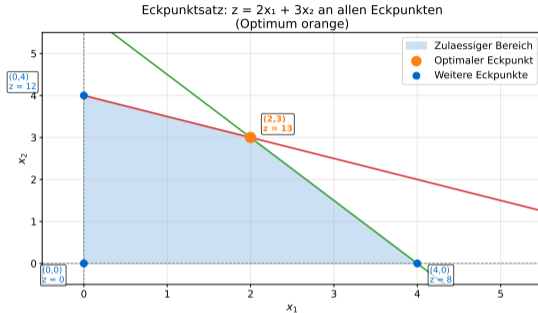
- Der **farbige Bereich** zeigt alle erlaubten Kombinationen
- **Ausserhalb** = geht nicht (zu wenig Mehl oder Zeit)
- Irgendwo **drin** liegt die beste Loesung

Alltags-Analogie

Wie eine Landkarte: Der schattierte Bereich ist das Gebiet, in dem Sie sich bewegen duerfen. Jede Grenzlinie ist eine Einschraenkung. Zusammen formen sie das erlaubte Gebiet.

Jede Grenzlinie ist eine Einschraenkung – zusammen formen sie das erlaubte Gebiet

Die Ueberraschung: Immer an einer Ecke!



Das Optimum liegt **IMMER** an einer Ecke!

Das ist die wichtigste Erkenntnis der linearen Optimierung.

Warum?

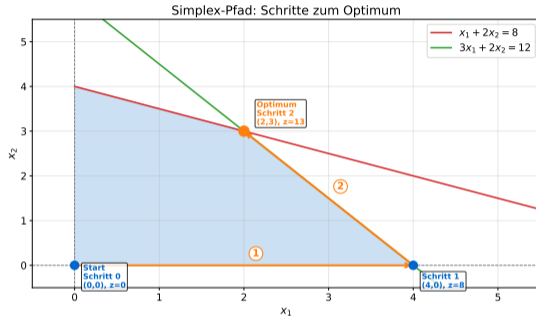
- Wenn Sie **im Inneren** stehen, koennen Sie immer noch ein Stueck weitergehen
- Erst an einer **Ecke** sind Sie wirklich "festgeklemmt"
- Kein weiterer Schritt verbessert das Ergebnis

Das ist **GROSSARTIG**:

Statt unendlich vieler Punkte pruefen wir nur die **wenigen Ecken**!

Der Eckpunktsatz ist die wichtigste Erkenntnis der linearen Optimierung

Wie findet man die beste Ecke?



Der Simplex-Algorithmus ist wie ein Bergsteiger

1. **Starte** an irgendeiner Ecke
2. **Schaue**: Ist eine Nachbar-Ecke besser?
3. **Wenn ja**: Gehe dorthin
4. **Wenn nein**: Du bist am Gipfel!

Einfach und genial

Man wandert **entlang der Kanten** des Polygons, immer bergauf, bis man die beste Ecke erreicht hat.

In der Praxis braucht der Algorithmus ueberraschend **wenige Schritte**.

George Dantzig erfand diesen Algorithmus 1947 – er revolutionierte die Wirtschaft

Was sind Schlupfvariablen?

Die Idee

Wenn Sie **6 von 8 kg** Mehl nutzen, bleiben **2 kg uebrig**.

Diese 2 kg heissen **Schlupf** – die ungenutzten Reste einer Ressource.

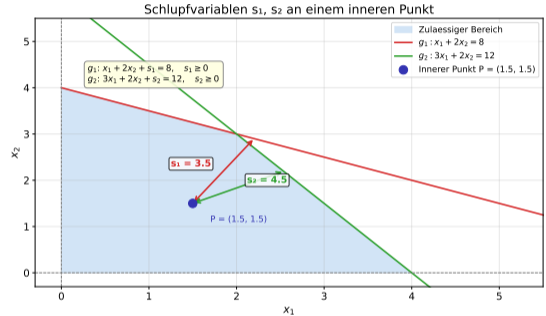
Der clevere Trick

Mehl verbraucht + Schlupf = Mehl vorhanden:

$$2x_1 + x_2 + s_1 = 8$$

Aus einer **Ungleichung** wird eine **Gleichung** – viel einfacher zu loesen!

Schlupfvariablen verwandeln Ungleichungen in Gleichungen – ein cleverer Trick



Wer benutzt das in der echten Welt?

Logistik

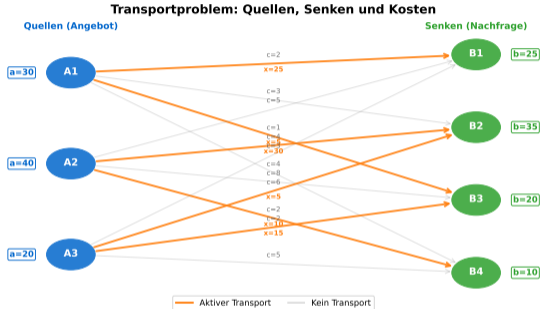
- Amazon, DHL optimieren **Millionen Lieferwege** taeglich

Finanzen

- Banken maximieren **Rendite** bei begrenztem Risiko

Produktion

- Fabriken planen **Maschinen, Material** und Personal



Milliarden-Euro-Entscheidungen werden taeglich mit LP getroffen!

Lineare Optimierung ist eines der meistgenutzten mathematischen Werkzeuge weltweit

5 Zeilen Python genuegen

```
from scipy.optimize import linprog  
c = [-3, -5] # Gewinn  
A = [[2,1],[1,2]] # Grenzen  
b = [8, 8]  
erg = linprog(c, A_ub=A, b_ub=b)
```

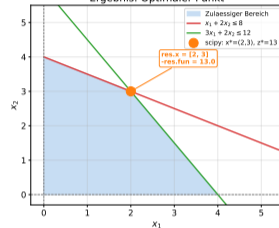
Warum negativ?

linprog **minimiert** immer.
Trick: Gewinn negieren, dann
minimieren = maximieren!

Python: LP loesen mit scipy

```
from scipy.optimize import linprog  
  
# max z = 2x1 + 3x2  
# => min -z = -2x1 - 3x2  
c = [-2, -3]  
  
# Ungleichungsrestriktionen (<=)  
A_ub = [[1, 2], [3, 2]]  
b_ub = [8, 12]  
  
# Variablen Grenzen x1, x2 >= 0  
bounds = [(0, None), (0, None)]  
  
res = linprog(c, A_ub=A_ub, b_ub=b_ub, bounds=bounds)  
  
print(res.x) # [2. 3.]  
print(-res.fun) # 13.0
```

Ergebnis: Optimaler Punkt



Warum Computer?

Python findet die Loesung in
Millisekunden – auch fuer
Probleme mit **tausenden Variablen!**

In der Praxis loest man LP nie von Hand – dafuer gibt es Computer

1. **LP = Das Beste aus begrenzten Mitteln**
2. **Erlaubter Bereich = Polygon aus Einschränkungen**
3. **Optimum liegt IMMER an einer Ecke**
4. **Simplex-Algorithmus: Von Ecke zu Ecke zum Gipfel**
5. **Python/SciPy löst es in Millisekunden**

Nächster Schritt: Selbst ein LP formulieren und lösen!

Merksatz

Lineare Optimierung:

Weniger ist nicht weniger –
es ist das **Beste**
aus dem Vorhandenen.

Ziel? Maximieren oder minimieren.

Trick? Nur die Ecken prüfen.

Werkzeug? Python erledigt den Rest.

Wir loesen die Baeckerei von Hand!

Unser Problem

- **Brote** (x_1): 3 EUR Gewinn
- **Kuchen** (x_2): 5 EUR Gewinn
- Mehl: $2x_1 + x_2 \leq 8$
- Zeit: $x_1 + 2x_2 \leq 8$
- Ziel: $z = 3x_1 + 5x_2$ maximieren

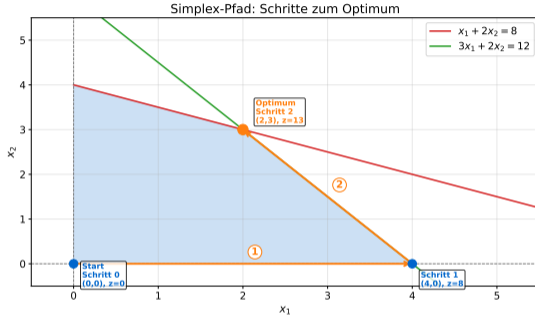
Die 4 Ecken

1. $(0, 0) \Rightarrow z = 0$
2. $(4, 0) \Rightarrow z = 12$
3. $(0, 4) \Rightarrow z = 20$
4. $(\frac{8}{3}, \frac{8}{3}) \Rightarrow z = \frac{64}{3} \approx 21,3$

\Rightarrow Ecke 4 gewinnt!

Der Simplex probiert NICHT alle Ecken – er wandert nur bergauf. Aber hier sind es nur 4.

Anhang: Simplex – Der Weg des Algorithmus



Schritt 1: Start bei $(0,0)$

Gewinn $z = 0$. Geht es besser?

→ **Ja!** Beide Richtungen verbessern z .

Schritt 2: Gehe zu $(0,4)$

Gewinn $z = 20$. Geht es besser?

→ **Ja!** Entlang der Kante nach rechts.

Schritt 3: Gehe zu $(\frac{8}{3}, \frac{8}{3})$

Gewinn $z \approx 21,3$. Geht es besser?

→ **Nein!** Alle Nachbarn sind schlechter.

OPTIMUM GEFUNDEN!

Nur 3 Schritte statt alle 4 Ecken prüfen – bei grossen Problemen spart das enorm viel Zeit

Der Simplex rechnet mit einer Tabelle (“Tableau”):

	x_1	x_2	s_1	s_2	Rechte Seite
s_1	2	1	1	0	8
s_2	1	2	0	1	8
z	-3	-5	0	0	0

Was steht in der Tabelle?

- Zeile 1: Mehl-Einschraenkung
- Zeile 2: Zeit-Einschraenkung
- Zeile 3: Zielfunktion (mit Minus!)
- s_1, s_2 : Schlupf (= Reserve)

Wie liest man die Loesung?

- Start: $x_1 = 0, x_2 = 0$
 - Schlupf: $s_1 = 8, s_2 = 8$
 - Gewinn: $z = 0$
- ⇒ Alles auf Reserve, nichts produziert

Der Simplex-Algorithmus veraendert diese Tabelle Schritt fuer Schritt, bis die optimale Ecke gefunden ist

Die grosse Sorge:

Das Problem

Der Simplex geht immer nur zum **besseren Nachbarn**.

Aber was, wenn das **globale Maximum** an einer Ecke liegt, die **KEIN Nachbar** ist?

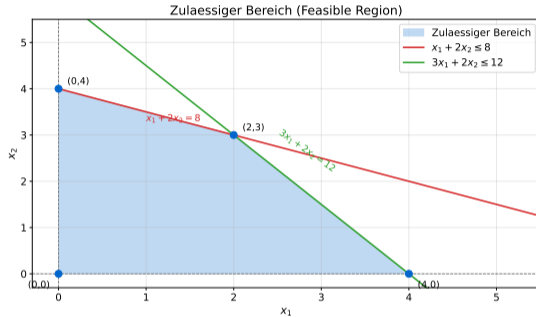
Verpassen wir die beste Loesung?

Vergleich: Bergsteigen

- Beim Wandern in den **Alpen** koennte man auf einem kleinen Huegel landen, waehrend der Mont Blanc woanders steht
 - Das waere ein **lokales Maximum**, nicht das **globale**
- ⇒ Bei Bergen: **JA, das passiert!**
- ⇒ Bei LP: **NEIN, das passiert NIE!**

Das ist DIE zentrale Frage – und die Antwort ist ueberraschend einfach

Warum es bei LP keine "versteckten Gipfel" gibt



Zwei magische Zutaten

- 1. Der Bereich ist KONVEX**
= keine Dellen, keine Loecher.
Zwischen zwei erlaubten Punkten sind **ALLE** Punkte dazwischen auch erlaubt.
- 2. Die Zielfunktion ist LINEAR**
= eine flache, gleichmaessig geneigte Ebene.
Keine Huegel, keine Taeler – nur eine einzige Richtung "bergauf".

Die Konsequenz

Konvex + Linear = Kein Versteck moeglich!

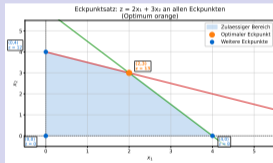
Wenn alle Nachbar-Ecken schlechter sind,
dann sind **ALLE** Ecken schlechter.

Konvexitaet + Linearitaet garantieren: jedes lokale Optimum ist automatisch das globale Optimum

Die Intuition: Warum "bergauf zum Nachbarn" reicht

Stellen Sie sich eine schiefe Tischplatte auf einem Polygon vor:

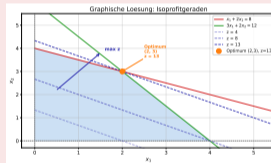
Berge (nichtlinear)



Viele Huegel \Rightarrow
man kann auf dem
falschen Huegel
haengenableiben

Lokal \neq Global

LP (linear)



Flache Ebene \Rightarrow
es gibt nur **EINE**
Richtung bergauf,
und die fuehrt zur Ecke

Lokal = Global!

Merksatz

**Bei linearer
Optimierung:**

Wenn kein Nachbar
besser ist, dann ist
NIEMAND besser.

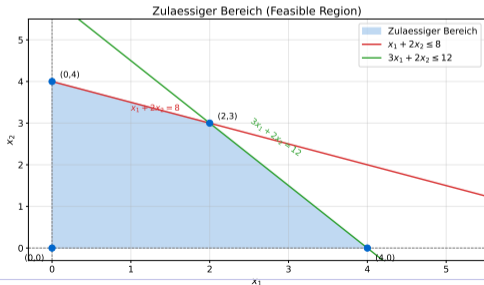
**Garantiert.
Immer.
Bewiesen.**

Deshalb funktioniert der Simplex so zuverlaessig – seit 1947 hat er noch nie das Optimum verpasst!

Die Idee: Warum brauchen wir ueberhaupt eine Tabelle?

Bei 2 Variablen: Zeichnen!

- Mit 2 Produkten (x_1, x_2) kann man den zulaessigen Bereich **zeichnen**
- Man **sieht** die Ecken
- Man **sieht**, welche Ecke die beste ist



Die Tabelle ist nur ein Werkzeug, damit der Computer die Ecken findet – die Idee bleibt: von Ecke zu Ecke bergauf

Bei 100 Variablen: Unmoeglich!

- Mit 100 Produkten kann man **nichts** zeichnen
 - Man braucht einen **systematischen Weg**, um von Ecke zu Ecke zu gehen
- ⇒ **Das ist die Tabelle!**

Die Tabelle ist eine Landkarte

Jede Tabelle beschreibt **eine Ecke**.

Wenn wir die Tabelle aendern,
wandern wir zur naechsten Ecke.

Das nennt man **“Pivotieren”**.

Unsere Baeckerei als Tabelle – wir starten bei Ecke (0, 0):

	x_1	x_2	s_1	s_2	RHS	Was bedeutet das?
s_1	2	1	1	0	8	8 kg Mehl uebrig (alles Reserve)
s_2	1	2	0	1	8	8 Std Zeit uebrig (alles Reserve)
z	-3	-5	0	0	0	Gewinn = 0 (wir machen ja nichts)

Was ist s_1 und s_2 ?

Schlupf = was uebrig bleibt.

$s_1 = 8$: Alle 8 kg Mehl sind uebrig, weil wir nichts produzieren.

$s_2 = 8$: Alle 8 Stunden sind uebrig.

Loesung ablesen: Variablen in der linken Spalte (BV) = ihr Wert steht rechts. Alle anderen = 0.

Was bedeuten -3 und -5?

Die Zahlen in der z-Zeile sagen:

“Pro Brot gewinnen wir 3 EUR, pro Kuchen 5 EUR.”

Negativ = es lohnt sich, das zu produzieren!

Schritt 1: Welches Produkt lohnt sich am meisten?

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

Regel: Schau in die unterste Zeile

Die z -Zeile hat zwei negative Zahlen:

- 3 bei x_1 (Brote bringen 3 EUR)
- 5 bei x_2 (Kuchen bringen 5 EUR)

-5 ist "am negativsten"

⇒ **Kuchen lohnt sich am meisten!**

⇒ **Spalte x_2 wird gewaehlt.**

Schritt 1 beantwortet: "Was sollen wir produzieren?" – Antwort: Das, was am meisten Gewinn bringt.

Warum die negativste?

Eine negative Zahl in der z -Zeile heisst: "*Dieses Produkt wuerde den Gewinn erhoehen.*"

Je negativer, desto mehr Gewinn pro Stueck.

-5 ist besser als -3

⇒ **Kuchen zuerst!**

Schritt 2: Wie viele Kuchen koennen wir maximal backen?

	x_1	x_2	s_1	s_2	RHS	
s_1	2	1	1	0	8	$8 \div 1 = 8$ Kuchen (Mehl reicht fuer 8)
s_2	1	2	0	1	8	$8 \div 2 = 4$ Kuchen (Zeit reicht fuer 4) ✓
z	-3	-5	0	0	0	

Regel: Teile RHS durch die Pivot-Spalte

Pro Kuchen brauchen wir:

1 kg Mehl $\Rightarrow 8 \div 1 = 8$ Kuchen moeglich

2 Std Zeit $\Rightarrow 8 \div 2 = 4$ Kuchen moeglich

4 ist kleiner als 8

\Rightarrow Die **Zeit** begrenzt uns zuerst!

\Rightarrow Maximal **4 Kuchen**.

Warum der kleinste?

Wir muessen **ALLE**
Einschraenkungen einhalten.

Das Mehl reicht fuer 8 Kuchen,
aber die Zeit nur fuer 4.

Der Engpass gewinnt!

(Wenn wir 8 backen, haetten
wir keine Zeit – ungueltig!)

Schritt 2 beantwortet: "Wie viel koennen wir produzieren?" – Antwort: So viel, bis der Engpass erreicht ist.

Schritt 3: Die Tabelle umbauen ("Pivotieren")

Was passiert beim Pivotieren?

Wir aendern die Tabelle so, dass sie die **neue Ecke** beschreibt.

Vorher: $x_2 = 0$ (keine Kuchen)

Nachher: $x_2 = 4$ (vier Kuchen)

Dafuer tauschen x_2 und s_2 :

- s_2 war "aktiv" (Zeit uebrig)
- Jetzt ist x_2 "aktiv" (Kuchen werden gebacken)
- $s_2 = 0$: keine Zeit mehr uebrig!

Die Rechenregel

1. Pivot-Zeile \div Pivot-Element (hier $\div 2$)
2. Andere Zeilen: so anpassen, dass in der Pivot-Spalte ueberall 0 steht

Wir sind von Ecke (0, 0) zu Ecke (0, 4) gewandert. Gewinn: 0 \rightarrow 20. Aber sind wir schon am Optimum?

Neue Tabelle = neue Ecke:

	x_1	x_2	s_1	s_2	RHS
s_1	$\frac{3}{2}$	0	1	$-\frac{1}{2}$	4
x_2	$\frac{1}{2}$	1	0	$\frac{1}{2}$	4
z	$-\frac{1}{2}$	0	0	$\frac{5}{2}$	20

Neue Loesung ablesen

$x_2 = 4$ Kuchen, $x_1 = 0$ Brote

$s_1 = 4$ (noch 4 kg Mehl uebrig)

$s_2 = 0$ (keine Zeit mehr!)

Gewinn: $z = 20$ EUR! (vorher 0)

Sind wir schon am Optimum? – Nochmal pruefen!

	x_1	x_2	s_1	s_2	RHS	
s_1	$\frac{3}{2}$	0	1	$-\frac{1}{2}$	4	$4 \div \frac{3}{2} = \frac{8}{3} \approx 2,67 \checkmark$
x_2	$\frac{1}{2}$	1	0	$\frac{1}{2}$	4	$4 \div \frac{1}{2} = 8$
z	$-\frac{1}{2}$	0	0	$\frac{5}{2}$	20	

Check: Noch negative in z-Zeile?

z-Zeile: $-\frac{1}{2}, 0, 0, \frac{5}{2}$

Ja! Die $-\frac{1}{2}$ bei x_1 .

Das heisst: *“Auch Brote wuerden den Gewinn noch erhoehen!”*

⇒ **Noch NICHT am Optimum.**

⇒ **Einen Pivot weiter!**

Die Frage “Sind wir fertig?” beantwortet immer die z-Zeile: **Negativ = es geht noch besser. Alles ≥ 0 = FERTIG.**

Gleiche Schritte nochmal:

1. **Spalte:** x_1 (einzige negative)
= Brote lohnen sich auch noch

2. **Quotienten:**
 $4 \div \frac{3}{2} = \frac{8}{3} \approx 2,67 \checkmark$
 $4 \div \frac{1}{2} = 8$
= Mehl begrenzt uns zuerst

3. **Pivotieren** auf $\frac{3}{2}$

Nach dem zweiten Pivot – die finale Tabelle:

	x_1	x_2	s_1	s_2	RHS	Bedeutung
x_1	1	0	$\frac{2}{3}$	$-\frac{1}{3}$	$\frac{8}{3}$	$\approx 2,67$ Brote
x_2	0	1	$-\frac{1}{3}$	$\frac{2}{3}$	$\frac{8}{3}$	$\approx 2,67$ Kuchen
z	0	0	$\frac{1}{3}$	$\frac{7}{3}$	$\frac{64}{3}$	$\approx 21,33$ EUR Gewinn

Warum sind wir jetzt fertig?

z-Zeile: 0, 0, $\frac{1}{3}$, $\frac{7}{3}$

Keine einzige negative Zahl!

Das heisst: *“Kein Produkt wuerde den Gewinn noch verbessern.”*

Mehr Brote? → Kein Platz.

Mehr Kuchen? → Kein Platz.

Die optimale Loesung

$\approx 2,67$ Brote und $\approx 2,67$ Kuchen

$z \approx 21,33$ EUR

Alle Ressourcen sind aufgebraucht:

$s_1 = 0$ (kein Mehl uebrig)

$s_2 = 0$ (keine Zeit uebrig)

Das Rezept

- 1. Tabelle aufschreiben**
LP mit Schlupfvariablen. Start: alles auf Null.
- 2. z-Zeile prüfen: Negative Zahl?**
Nein → FERTIG! Ja → weiter:
- 3. Spalte wählen** (negativste in z-Zeile)
= "Was lohnt sich am meisten?"
- 4. Zeile wählen** (kleinster Quotient RHS/Spalte)
= "Wo ist der Engpass?"
- 5. Pivotieren** (Zeilen umrechnen)
= "Zur nächsten Ecke wandern"
- 6. Zurück zu Schritt 2**

Unser Weg

Ecke 1: $(0, 0)$
 $z = 0$ EUR

↓ Kuchen lohnt sich → Pivot

Ecke 2: $(0, 4)$
 $z = 20$ EUR

↓ Brote lohnen sich auch → Pivot

Ecke 3: $(\frac{8}{3}, \frac{8}{3})$
 $z \approx 21,33$ EUR

Nichts lohnt sich mehr → **STOPP!**

Jede Tabelle = eine Ecke.
Jeder Pivot = bergauf.
Keine Negative = Gipfel.