

Vorlesung 10: Mehrfache Zielsetzung

Grundlagen Operations Research

Prof. Dr. Knut Haase

Mehrere Ziele

Zwei Ziele, ein Plan

Erinnerung: das LKW-Beispiel

Auftrag	Stückerlös	Paletten	Tonnen
1	20	1	1
2	30	1	3
Kapazität		5	9

Für das Be- und Entladen fallen **5 EUR pro Tonne** an.

Ziel 1: Umsatz

$$\max F = 20X_1 + 30X_2$$

$$X_1 + X_2 \leq 5, \quad X_1 + 3X_2 \leq 9, \quad X_1, X_2 \geq 0$$

Ziel 1: Umsatz

$$\max F = 20X_1 + 30X_2$$

$$X_1 + X_2 \leq 5, \quad X_1 + 3X_2 \leq 9, \quad X_1, X_2 \geq 0$$

$$X_1^* = 3, \quad X_2^* = 2, \quad F^* = 120$$

Deckungsbeiträge

Zieht man die **5 EUR/Tonne** ab, ergibt sich je Palette:

$$DB_1 = 20 - 1 \cdot 5 = 15, \quad DB_2 = 30 - 3 \cdot 5 = 15$$

Deckungsbeiträge

Zieht man die **5 EUR/Tonne** ab, ergibt sich je Palette:

$$DB_1 = 20 - 1 \cdot 5 = 15, \quad DB_2 = 30 - 3 \cdot 5 = 15$$

Das Controlling empfiehlt, nach **Deckungsbeitrag** statt Umsatz zu planen.

Ziel 2: Deckungsbeitrag

$$\max F = 15X_1 + 15X_2$$

$$X_1 + X_2 \leq 5, \quad X_1 + 3X_2 \leq 9, \quad X_1, X_2 \geq 0$$

Ziel 2: Deckungsbeitrag

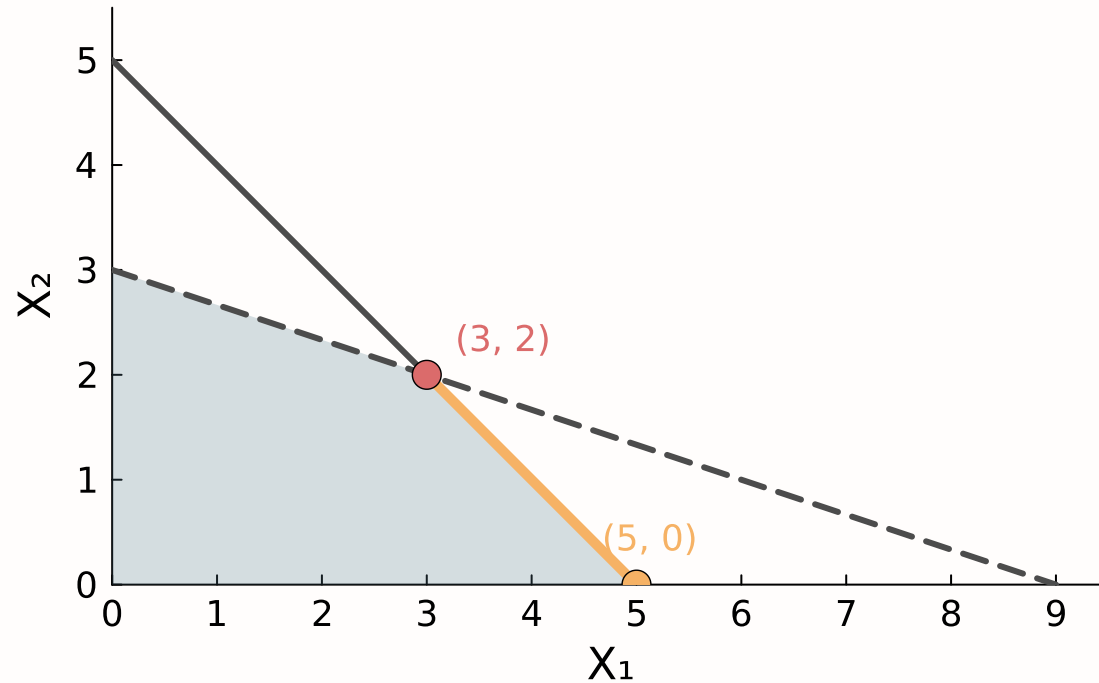
$$\max F = 15X_1 + 15X_2$$

$$X_1 + X_2 \leq 5, \quad X_1 + 3X_2 \leq 9, \quad X_1, X_2 \geq 0$$

$$X_1^* = 5, \quad X_2^* = 0, \quad F^* = 75$$

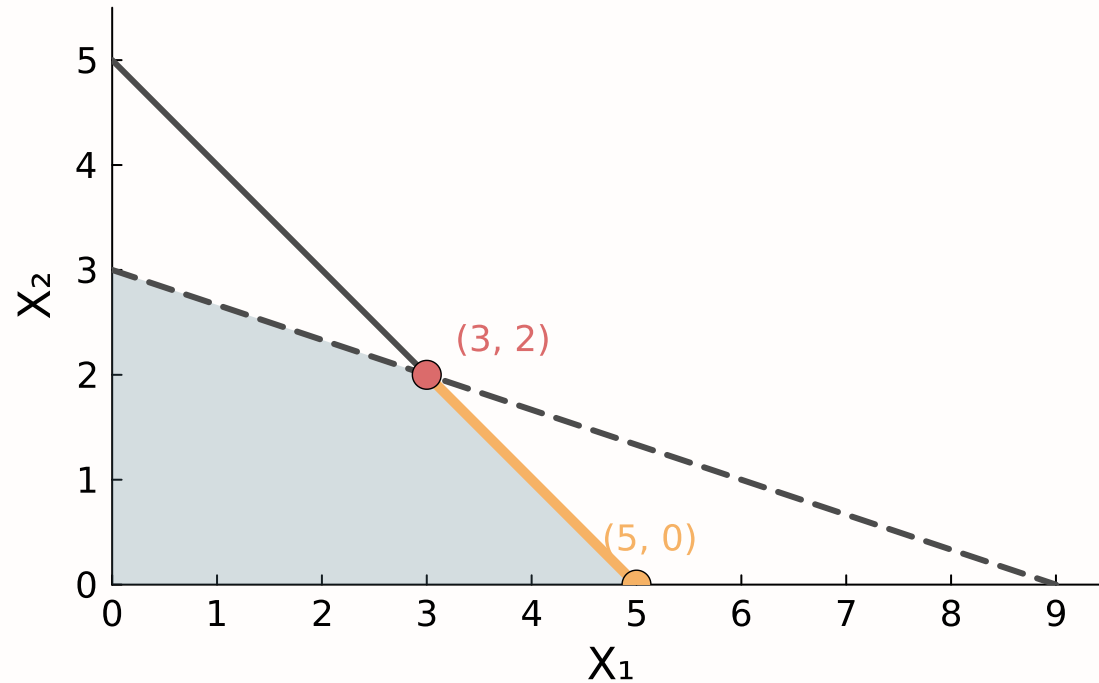
Die Umsatzerlöse sinken dabei auf $5 \cdot 20 = 100$ EUR und die ungenutzte Ladekapazität steigt auf 4 Tonnen: ein **Zielkonflikt**.

Der Zielkonflikt



- **Umsatz-Optimum:** (3, 2) (rot).
- **Deckungsbeitrag-Optimum:** (5, 0) (orange).

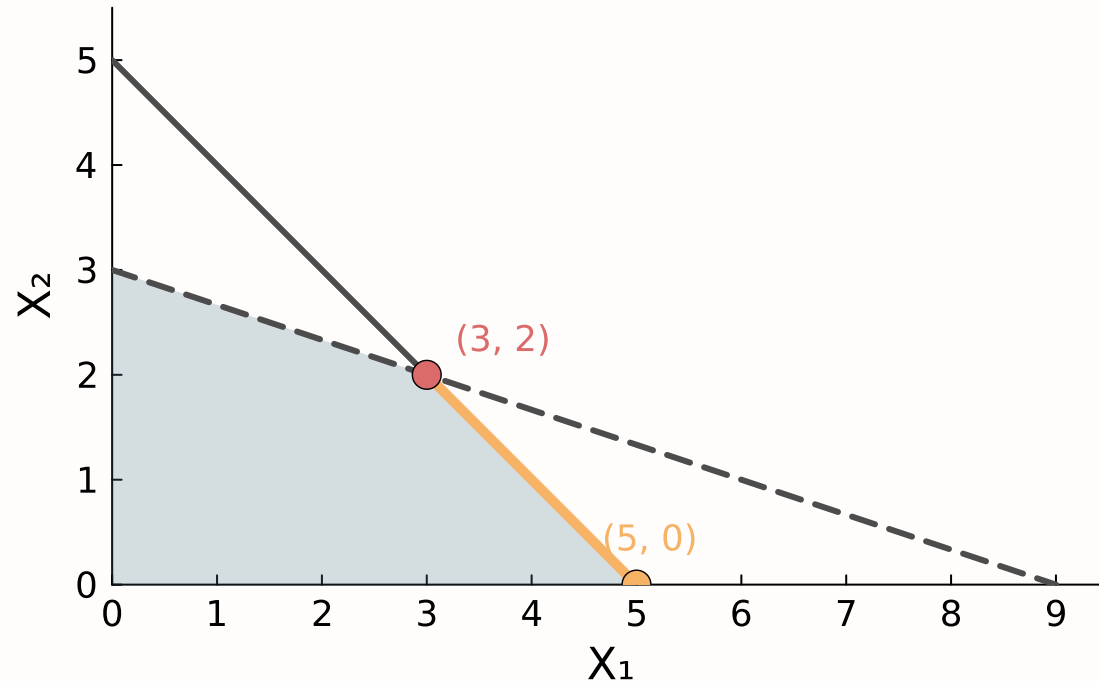
Der Zielkonflikt



- **Umsatz-Optimum:** (3,2) (rot).
- **Deckungsbeitrag-Optimum:** (5,0) (orange).

Die **orangefarbene Kante** ist komplett DB-optimal: mehrdeutig (V05!).

Der Zielkonflikt



- **Umsatz-Optimum:** (3, 2) (rot).
- **Deckungsbeitrag-Optimum:** (5, 0) (orange).

Die **orangefarbene Kante** ist komplett DB-optimal: mehrdeutig (V05!).

Frage: Wie lösen wir den Konflikt auf?

Arten von Zielkonflikten

Arten von Zielkonflikten

- **Konkurrierende** (konträre) Ziele: sie behindern sich.

Arten von Zielkonflikten

- **Konkurrierende** (konträre) Ziele: sie behindern sich.
- **Komplementäre** Ziele: sie unterstützen sich.

Arten von Zielkonflikten

- **Konkurrierende** (konträre) Ziele: sie behindern sich.
- **Komplementäre** Ziele: sie unterstützen sich.
- **Neutrale** (orthogonale) Ziele: sie beeinflussen sich nicht.

Arten von Zielkonflikten

- **Konkurrierende** (konträre) Ziele: sie behindern sich.
- **Komplementäre** Ziele: sie unterstützen sich.
- **Neutrale** (orthogonale) Ziele: sie beeinflussen sich nicht.

Vier Ansätze zur Auflösung: **lexikographische Ordnung**, **Haupt-/Nebenziele**, **Zielgewichtung**, **Goal Programming**. Maßstab: der **Zielerreichungsgrad**.

Lexikographisch

Ziele nach Wichtigkeit

Eine strikte Rangfolge der Ziele $A > B > C > \dots$

Ziele nach Wichtigkeit

Eine strikte Rangfolge der Ziele $A > B > C > \dots$

1. Optimiere **nur** Ziel $A \Rightarrow$ Lösungsmenge X_A .

Ziele nach Wichtigkeit

Eine strikte Rangfolge der Ziele $A > B > C > \dots$

1. Optimierte **nur** Ziel $A \Rightarrow$ Lösungsmenge X_A .

2. Optimierte Ziel B **nur auf** $X_A \Rightarrow X_B \subseteq X_A$.

Ziele nach Wichtigkeit

Eine strikte Rangfolge der Ziele $A > B > C > \dots$

1. Optimiere **nur** Ziel $A \Rightarrow$ Lösungsmenge X_A .
2. Optimiere Ziel B **nur auf** $X_A \Rightarrow X_B \subseteq X_A$.
3. Optimiere Ziel C **nur auf** X_B , und so weiter.

Anwendung auf das Beispiel

Ziel A (Deckungsbeitrag): $F_A = 75$. Optimal ist die **ganze Kante** von $(5, 0)$ bis $(3, 2)$, also gibt es noch Spielraum.

Anwendung auf das Beispiel

Ziel A (Deckungsbeitrag): $F_A = 75$. Optimal ist die **ganze Kante** von $(5, 0)$ bis $(3, 2)$, also gibt es noch Spielraum.

Ziel B (Umsatz), aber **Deckungsbeitrag beibehalten**:

$$\max F_B = 20X_1 + 30X_2$$

$$15X_1 + 15X_2 \geq 75 \quad (\text{DB-Niveau halten})$$

$$X_1 + X_2 \leq 5, \quad X_1 + 3X_2 \leq 9$$

Anwendung auf das Beispiel

Ziel A (Deckungsbeitrag): $F_A = 75$. Optimal ist die **ganze Kante** von $(5, 0)$ bis $(3, 2)$, also gibt es noch Spielraum.

Ziel B (Umsatz), aber **Deckungsbeitrag beibehalten**:

$$\max F_B = 20X_1 + 30X_2$$

$$15X_1 + 15X_2 \geq 75 \quad (\text{DB-Niveau halten})$$

$$X_1 + X_2 \leq 5, \quad X_1 + 3X_2 \leq 9$$

$\Rightarrow X_1^* = 3, X_2^* = 2, F_B^* = 120$: beide Ziele erfüllt.

Haupt-/Nebenziel

Hauptziel plus Nebenziel

Ein Ziel wird **optimiert**, die übrigen als **Mindestanforderung** (satisfizierend) formuliert.

$$\max F = 15X_1 + 15X_2$$

(Hauptziel: DB)

$$20X_1 + 30X_2 \geq 110$$

(Nebenziel: Umsatz)

$$X_1 + X_2 \leq 5, \quad X_1 + 3X_2 \leq 9$$

Hauptziel plus Nebenziel

Ein Ziel wird **optimiert**, die übrigen als **Mindestanforderung** (satisfizierend) formuliert.

$$\max F = 15X_1 + 15X_2$$

(Hauptziel: DB)

$$20X_1 + 30X_2 \geq 110$$

(Nebenziel: Umsatz)

$$X_1 + X_2 \leq 5, \quad X_1 + 3X_2 \leq 9$$

$$X_1^* = 3, \quad X_2^* = 2, \quad F^* = 75$$

Hauptziel plus Nebenziel

Ein Ziel wird **optimiert**, die übrigen als **Mindestanforderung** (satisfizierend) formuliert.

$$\max F = 15X_1 + 15X_2$$

(Hauptziel: DB)

$$20X_1 + 30X_2 \geq 110$$

(Nebenziel: Umsatz)

$$X_1 + X_2 \leq 5, \quad X_1 + 3X_2 \leq 9$$

$$X_1^* = 3, \quad X_2^* = 2, \quad F^* = 75$$

Mehrdeutig: Jeder Punkt der Kante von (3, 2) bis (4, 1) hat $F = 75$ und mindestens 110 EUR Umsatz.

Zielgewichtung

Gewichtete Summe

Alle Ziele **gleichrichten** (Minimierung $\times(-1)$) und mit Gewichten $0 \leq \lambda_i \leq 1$ zusammenfassen:

$$\text{Maximiere } Z = \sum_{i=1}^t \lambda_i F_i$$

Gewichtete Summe

Alle Ziele **gleichrichten** (Minimierung $\times(-1)$) und mit Gewichten $0 \leq \lambda_i \leq 1$ zusammenfassen:

$$\text{Maximiere } Z = \sum_{i=1}^t \lambda_i F_i$$

Die Gewichte drücken die **relative Bedeutung** der Ziele aus.

Beispiel

Umsatz und Deckungsbeitrag **gleich** gewichtet, $\lambda_1 = \lambda_2 = 0,5$:

$$Z = 0,5 (20X_1 + 30X_2) + 0,5 (15X_1 + 15X_2) = 17,5X_1 + 22,5X_2$$

Beispiel

Umsatz und Deckungsbeitrag **gleich** gewichtet, $\lambda_1 = \lambda_2 = 0,5$:

$$Z = 0,5 (20X_1 + 30X_2) + 0,5 (15X_1 + 15X_2) = 17,5X_1 + 22,5X_2$$

$$X_1^* = 3, \quad X_2^* = 2, \quad Z^* = 97,5$$

Goal Programming

Abstand zum Ideal

Erweiterung der Zielgewichtung: Minimiere den **Abstand** zwischen dem Ideal F_i^* und dem erreichten Wert F_i .

Abstand zum Ideal

Erweiterung der Zielgewichtung: Minimiere den **Abstand** zwischen dem Ideal F_i^* und dem erreichten Wert F_i .

- Summe der gewichteten Abstände: $Z_1 = \sum_i \lambda_i |F_i^* - F_i|$.

Abstand zum Ideal

Erweiterung der Zielgewichtung: Minimiere den **Abstand** zwischen dem Ideal F_i^* und dem erreichten Wert F_i .

- Summe der gewichteten Abstände: $Z_1 = \sum_i \lambda_i |F_i^* - F_i|$.
- Maximaler gewichteter Abstand: $Z_2 = \max_i \lambda_i |F_i^* - F_i|$.

Abstand zum Ideal

Erweiterung der Zielgewichtung: Minimiere den **Abstand** zwischen dem Ideal F_i^* und dem erreichten Wert F_i .

- Summe der gewichteten Abstände: $Z_1 = \sum_i \lambda_i |F_i^* - F_i|$.
- Maximaler gewichteter Abstand: $Z_2 = \max_i \lambda_i |F_i^* - F_i|$.
- Summe der quadrierten Abstände: $Z_3 = \sum_i \lambda_i (F_i^* - F_i)^2$.

Ohne Betragsstriche

Frage: Warum dürfen die Beträge einfach wegfallen?

Ohne Betragsstriche

Frage: Warum dürfen die Beträge einfach wegfallen?

Im zulässigen Bereich gilt für Maximierungsziele stets $F_i \leq F_i^*$: der Abstand $F_i^* - F_i$ ist nie negativ.

Ohne Betragsstriche

Frage: Warum dürfen die Beträge einfach wegfallen?

Im zulässigen Bereich gilt für Maximierungsziele stets $F_i \leq F_i^*$: der Abstand $F_i^* - F_i$ ist nie negativ.

Für Z_1 (mit $\lambda_i = 1$) heißt das konkret:

$$\begin{aligned} Z_1 &= (75 - 15X_1 - 15X_2) + (120 - 20X_1 - 30X_2) \rightarrow \min \\ &\Leftrightarrow \max 35X_1 + 45X_2 \end{aligned}$$

Ohne Betragsstriche

Frage: Warum dürfen die Beträge einfach wegfallen?

Im zulässigen Bereich gilt für Maximierungsziele stets $F_i \leq F_i^*$: der Abstand $F_i^* - F_i$ ist nie negativ.

Für Z_1 (mit $\lambda_i = 1$) heißt das konkret:

$$\begin{aligned} Z_1 &= (75 - 15X_1 - 15X_2) + (120 - 20X_1 - 30X_2) \rightarrow \min \\ &\Leftrightarrow \max 35X_1 + 45X_2 \end{aligned}$$

Wieder ein gewöhnliches LP: Optimum (3, 2) mit $Z_1 = 0$.

Der Min-Max-Ansatz

Je Ziel eine Restriktion, $\lambda_1 = \lambda_2 = 1$:

$\min Z$

$$Z + 20X_1 + 30X_2 \geq 120 \quad (\text{Abstand Umsatz})$$

$$Z + 15X_1 + 15X_2 \geq 75 \quad (\text{Abstand DB})$$

$$X_1 + X_2 \leq 5, \quad X_1 + 3X_2 \leq 9$$

Der Min-Max-Ansatz

Je Ziel eine Restriktion, $\lambda_1 = \lambda_2 = 1$:

min Z

$$Z + 20X_1 + 30X_2 \geq 120 \quad (\text{Abstand Umsatz})$$

$$Z + 15X_1 + 15X_2 \geq 75 \quad (\text{Abstand DB})$$

$$X_1 + X_2 \leq 5, \quad X_1 + 3X_2 \leq 9$$

$$X_1^* = 3, \quad X_2^* = 2, \quad Z^* = 0$$

Der Min-Max-Ansatz

Je Ziel eine Restriktion, $\lambda_1 = \lambda_2 = 1$:

min Z

$$Z + 20X_1 + 30X_2 \geq 120 \quad (\text{Abstand Umsatz})$$

$$Z + 15X_1 + 15X_2 \geq 75 \quad (\text{Abstand DB})$$

$$X_1 + X_2 \leq 5, \quad X_1 + 3X_2 \leq 9$$

$$X_1^* = 3, \quad X_2^* = 2, \quad Z^* = 0$$

! Kein echter Zielkonflikt

Der Punkt (3,2) erreicht **beide** Ideale (Umsatz 120, DB 75). Die Ziele konkurrieren in diesem Beispiel also nicht wirklich, deshalb ist (3,2) bei **allen** Ansätzen optimal.

Echter Konflikt

Variante: 8 EUR pro Tonne

Das Be- und Entladen kostet jetzt 8 EUR pro Tonne statt 5:

$$DB_1 = 20 - 1 \cdot 8 = 12, \quad DB_2 = 30 - 3 \cdot 8 = 6$$

Variante: 8 EUR pro Tonne

Das Be- und Entladen kostet jetzt **8 EUR pro Tonne** statt 5:

$$DB_1 = 20 - 1 \cdot 8 = 12, \quad DB_2 = 30 - 3 \cdot 8 = 6$$

Ziel 2 bei unveränderten Restriktionen:

$$\max F = 12X_1 + 6X_2 \quad \Rightarrow \quad X_1^* = 5, \quad X_2^* = 0, \quad F^* = 60$$

Variante: 8 EUR pro Tonne

Das Be- und Entladen kostet jetzt **8 EUR pro Tonne** statt 5:

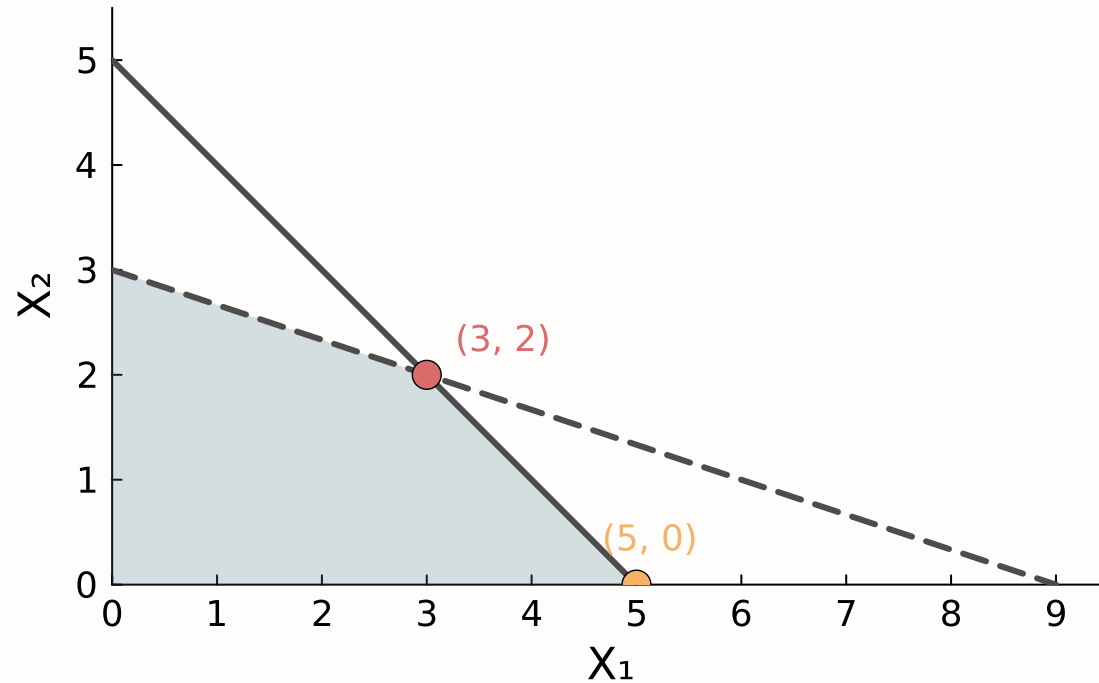
$$DB_1 = 20 - 1 \cdot 8 = 12, \quad DB_2 = 30 - 3 \cdot 8 = 6$$

Ziel 2 bei unveränderten Restriktionen:

$$\max F = 12X_1 + 6X_2 \quad \Rightarrow \quad X_1^* = 5, \quad X_2^* = 0, \quad F^* = 60$$

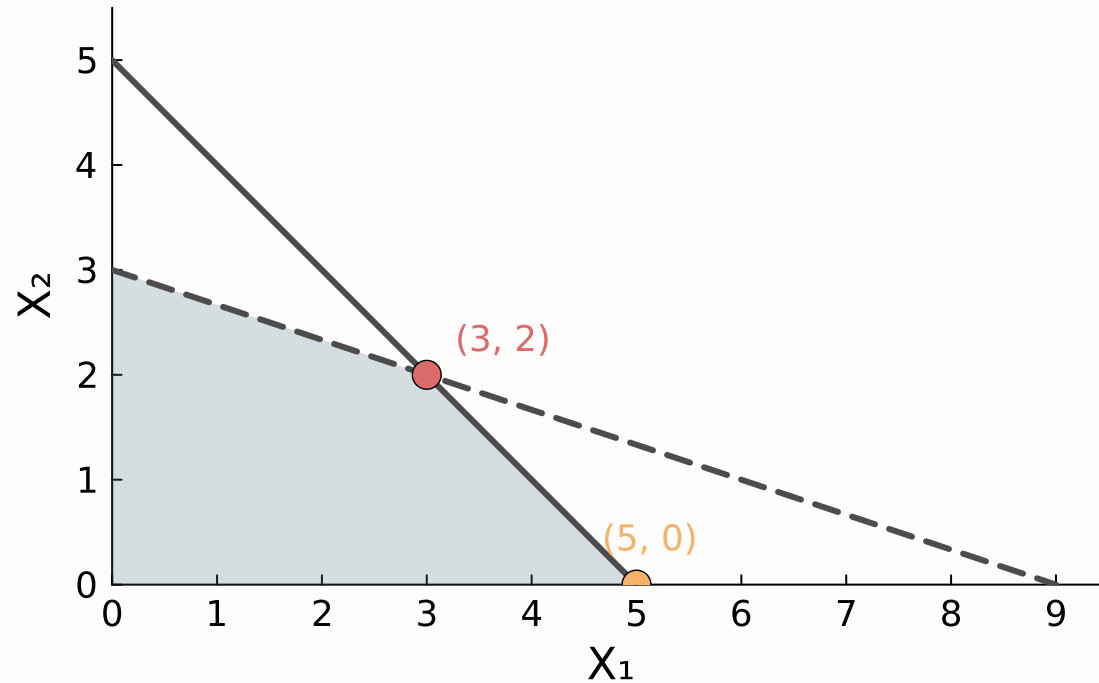
Dieses Optimum ist **eindeutig**, der Umsatz beträgt dort nur 100 EUR.

Kein gemeinsames Optimum



- **Umsatz**-Optimum (3, 2): DB nur 48.
- **DB**-Optimum (5, 0): Umsatz nur 100.

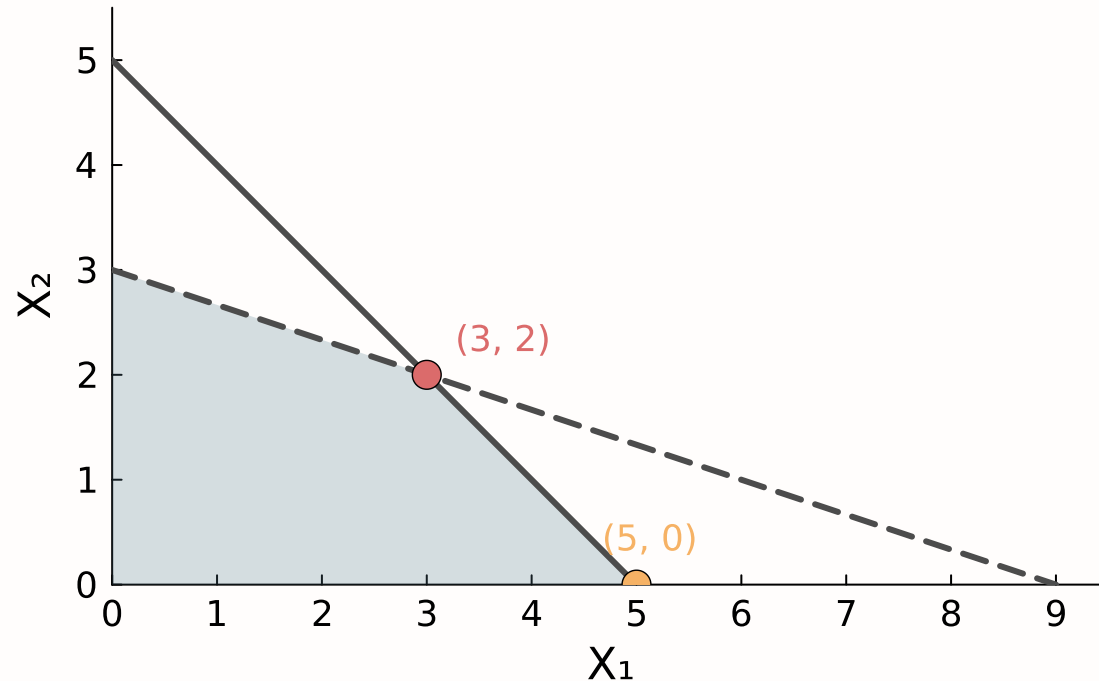
Kein gemeinsames Optimum



- **Umsatz**-Optimum (3, 2): DB nur 48.
- **DB**-Optimum (5, 0): Umsatz nur 100.

Kein Punkt erreicht **beide** Ideale (120 und 60): Die Ziele **konkurrieren**.

Kein gemeinsames Optimum



- **Umsatz**-Optimum (3, 2): DB nur 48.
- **DB**-Optimum (5, 0): Umsatz nur 100.

Kein Punkt erreicht **beide** Ideale (120 und 60): Die Ziele **konkurrieren**.

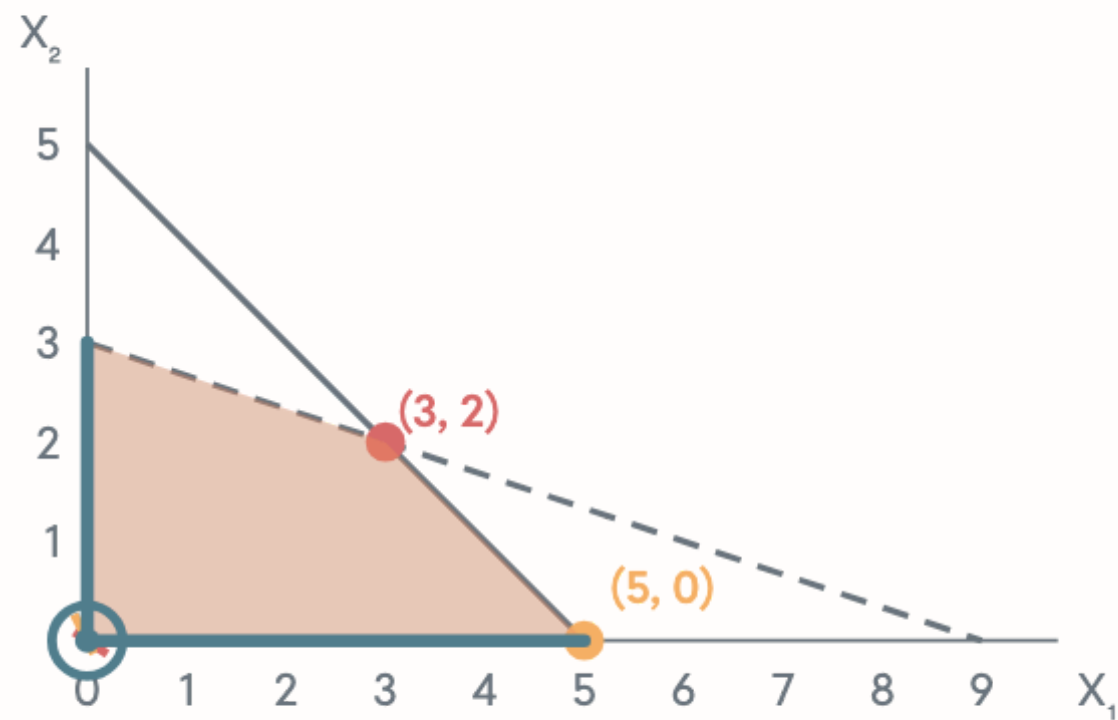


Gleich ausprobieren

Auf der nächsten Folie suchen Sie selbst den Min-Max-Kompromiss: den Plan, bei dem der größere der beiden Abstände zum Ideal am kleinsten ist. Liegt er an einer Ecke?

Sie sind dran

Umsatz und Deckungsbeitrag haben verschiedene Ideale. Ziehen Sie den Punkt auf den besten Plan nach dem angegebenen Ansatz. Getönt ist, wo ein Ziel besser wäre.



Goal Programming · min Z_2 (Min-Max)

Umsatz = $20 X_1 + 30 X_2$

DB = $12 X_1 + 6 X_2$

$X_1 = 0$	$X_2 = 0$		
	Wert	Ideal	Abstand
Umsatz	0	120	120
DB	0	60	60
$Z_2 = \text{größerer Abstand}$			120



Eigener Plan **120**

Bereit: „Optimieren“
vergleicht den Plan
mit dem besten.

Optimieren

Reset

Neu

Die Ansätze im Vergleich

Im Spiel: Min-Max-Kompromiss $(3,75; 1,25)$ auf der Kante, beide Abstände 7,5.

Ansatz	Lösung	Umsatz	DB
Lexikographisch	$(5,0)$	100	60
Haupt-/Nebenziele	$(4,1)$	110	54
Zielgewichtung	$(3,2)$	120	48
Goal Programming Z_1	$(3,2)$	120	48
Goal Programming Z_2	$(3,75; 1,25)$	112,5	52,5

Welcher Ansatz?

! Ergebnis

Dieselben Ansätze wie zuvor liefern jetzt **unterschiedliche** Empfehlungen. Bei einem echten Zielkonflikt müssen Sie den Ansatz selbst **wählen**.

Kontrolle mit Julia

Goal Programming mit JuMP

Zuerst die beiden Einzeloptima F_1^* , F_2^* , dann der Ausgleich:

```
1 using JuMP, HiGHS
2
3 # Einzeloptima bestimmen
4 ideal(c1, c2) = begin
5     modell = Model(HiGHS.Optimizer); set_silent(modell)
6     @variable(modell, X[1:2] >= 0)
7     @objective(modell, Max, c1*X[1] + c2*X[2])
8     @constraint(modell, X[1] + X[2] <= 5)
9     @constraint(modell, X[1] + 3X[2] <= 9)
10    optimize!(modell); objective_value(modell)
11 end
```

Min-Max-Ansatz mit JuMP

```
1 function ausgleich(db1, db2)           # minimiere max. Abweichung
2     F1, F2 = ideal(20, 30), ideal(db1, db2) # Umsatz-, DB-Maximum
3     modell = Model(HiGHS.Optimizer); set_silent(modell)
4     @variable(modell, X[1:2] >= 0); @variable(modell, Z >= 0)
5     @constraint(modell, Z + 20X[1] + 30X[2] >= F1)
6     @constraint(modell, Z + db1*X[1] + db2*X[2] >= F2)
7     @constraint(modell, X[1] + X[2] <= 5)
8     @constraint(modell, X[1] + 3X[2] <= 9)
9     @objective(modell, Min, Z)
10    optimize!(modell)
11    wert(v) = round(value(v), digits = 2)
12    println("X = ", wert.(X), ", Umsatz = ", wert(20X[1] + 30X[2]),
13           ", DB = ", wert(db1*X[1] + db2*X[2]), ", Z* = ", wert(Z))
14 end
15 ausgleich(15, 15)           # 5 EUR pro Tonne
16 ausgleich(12, 6)           # 8 EUR pro Tonne
```

Min-Max-Ansatz mit JuMP

```
1 function ausgleich(db1, db2)           # minimiere max. Abweichung
2     F1, F2 = ideal(20, 30), ideal(db1, db2) # Umsatz-, DB-Maximum
3     modell = Model(HiGHS.Optimizer); set_silent(modell)
4     @variable(modell, X[1:2] >= 0); @variable(modell, Z >= 0)
5     @constraint(modell, Z + 20X[1] + 30X[2] >= F1)
6     @constraint(modell, Z + db1*X[1] + db2*X[2] >= F2)
7     @constraint(modell, X[1] + X[2] <= 5)
8     @constraint(modell, X[1] + 3X[2] <= 9)
9     @objective(modell, Min, Z)
10    optimize!(modell)
11    wert(v) = round(value(v), digits = 2)
12    println("X = ", wert.(X), ", Umsatz = ", wert(20X[1] + 30X[2]),
13           ", DB = ", wert(db1*X[1] + db2*X[2]), ", Z* = ", wert(Z))
14 end
15 ausgleich(15, 15)           # 5 EUR pro Tonne
16 ausgleich(12, 6)           # 8 EUR pro Tonne
```

X = [3.0, 2.0], Umsatz = 120.0, DB = 75.0, Z* = 0.0

X = [3.75, 1.25], Umsatz = 112.5, DB = 52.5, Z* = 7.5

Standortplanung

Ein Gebietsplanungsproblem

16 Gebiete (4×4-Raster) mit Umsatzpotenzialen; **4 mögliche Standorte** (Gebiete 2, 8, 9, 15, grau). Wähle **genau zwei** und ordne jedes Gebiet einem Standort zu.

1 68	2 96	3 101	4 114
5 42	6 73	7 78	8 34
9 68	10 51	11 73	12 52
13 51	14 105	15 123	16 77

Variablen und Restriktionen

Frage: Welche Variablen und Restriktionen brauchen wir?

Variablen und Restriktionen

Frage: Welche Variablen und Restriktionen brauchen wir?

- $Y_j \in \{0, 1\}$: Standort j errichten; $X_{ij} \in \{0, 1\}$: Gebiet $i \rightarrow$ Standort j .

Variablen und Restriktionen

Frage: Welche Variablen und Restriktionen brauchen wir?

- $Y_j \in \{0, 1\}$: Standort j errichten; $X_{ij} \in \{0, 1\}$: Gebiet $i \rightarrow$ Standort j .
- Genau zwei Standorte: $\sum_j Y_j = 2$.

Zuordnung der Gebiete

Zuordnung der Gebiete

- Zuordnung nur zu **errichteten**: $X_{ij} \leq Y_j$; ein Standort versorgt sein **eigenes** Gebiet: $X_{jj} = Y_j$.

Zuordnung der Gebiete

- Zuordnung nur zu **errichteten**: $X_{ij} \leq Y_j$; ein Standort versorgt sein **eigenes** Gebiet: $X_{jj} = Y_j$.
- Jedes Gebiet **genau einmal**: $\sum_j X_{ij} = 1$.

Zuordnung der Gebiete

- Zuordnung nur zu **errichteten**: $X_{ij} \leq Y_j$; ein Standort versorgt sein **eigenes** Gebiet: $X_{jj} = Y_j$.
- Jedes Gebiet **genau einmal**: $\sum_j X_{ij} = 1$.

Im Original zusätzlich: Gebiete müssen **zusammenhängend** sein.
Das lassen wir hier weg.

Zwei konkurrierende Ziele

Fairness gegen Kosten

- F_1 = **kleinstes** zugeordnetes Umsatzpotenzial $\rightarrow$ **maximieren** (ausgewogene Gebiete).
- F_2 = gesamte **Reisekosten** $\sum_{ij} d_{ij} X_{ij} \rightarrow$ **minimieren**.

Zwei konkurrierende Ziele

Fairness gegen Kosten

- F_1 = **kleinstes** zugeordnetes Umsatzpotenzial $\rightarrow$ **maximieren** (ausgewogene Gebiete).
- F_2 = gesamte **Reisekosten** $\sum_{ij} d_{ij} X_{ij} \rightarrow$ **minimieren**.

Ein Zielkonflikt, aufzulösen mit **Goal Programming**
(gleichgewichtete **relative** Abweichungen).

Daten und Entfernungen

```
1 using JuMP, HiGHS
2 u = [68,96,101,114, 42,73,78,34, 68,51,73,52, 51,105,123,77]
3 J = [2,8,9,15]; I = 1:16
4 pos(i) = (cld(i,4), mod1(i,4)) # Zeile, Spalte
5 d(i,j) = abs(pos(i)[1]-pos(j)[1]) + abs(pos(i)[2]-pos(j)[2])
```



Das Grundmodell

```
1 function standortmodell()
2     modell = Model(HiGHS.Optimizer); set_silent(modell)
3     @variable(modell, Y[J], Bin); @variable(modell, X[I,J], Bin)
4     @variable(modell, F1 >= 0)
5     @constraint(modell, sum(Y[j] for j in J) == 2)
6     @constraint(modell, [i in I, j in J], X[i,j] <= Y[j])
7     @constraint(modell, [j in J], X[j,j] == Y[j])
8     @constraint(modell, [i in I], sum(X[i,j] for j in J) == 1)
9     @constraint(modell, [j in J],
10         F1 <= sum(u[i]*X[i,j] for i in I) + sum(u)*(1 - Y[j]))
11     modell, X, Y, F1, sum(d(i,j)*X[i,j] for i in I, j in J)
12 end
```

Die Einzeloptima

Erst die beiden Ideale, jeweils mit demselben Grundmodell:

```
1 m1, _, _, F1v, _ = standortmodell()
2 @objective(m1, Max, F1v); optimize!(m1); F1stern = objective_value(m1)
3 m2, _, _, _, F2v = standortmodell()
4 @objective(m2, Min, F2v); optimize!(m2); F2stern = objective_value(m2)
5 println("F1* = ", round{Int, F1stern},
6         ", F2* = ", round{Int, F2stern})
```

Die Einzeloptima

Erst die beiden Ideale, jeweils mit demselben Grundmodell:

```
1 m1, _, _, F1v, _ = standortmodell()
2 @objective(m1, Max, F1v); optimize!(m1); F1stern = objective_value(m1)
3 m2, _, _, _, F2v = standortmodell()
4 @objective(m2, Min, F2v); optimize!(m2); F2stern = objective_value(m2)
5 println("F1* = ", round{Int, F1stern},
6         ", F2* = ", round{Int, F2stern})
```

F1* = 603, F2* = 24

Die Ideale auf der Karte

Fair: Potenziale 603 | 603,
Reisekosten 41

1 68	2 96	3 101	4 114
5 42	6 73	7 78	8 34
9 68	10 51	11 73	12 52
13 51	14 105	15 123	16 77

Die Ideale auf der Karte

Fair: Potenziale 603 | 603,
Reisekosten 41

1 68	2 96	3 101	4 114
5 42	6 73	7 78	8 34
9 68	10 51	11 73	12 52
13 51	14 105	15 123	16 77

Kurz: Potenziale 625 | 581,
Reisekosten 24

1 68	2 96	3 101	4 114
5 42	6 73	7 78	8 34
9 68	10 51	11 73	12 52
13 51	14 105	15 123	16 77

Die Ideale auf der Karte

Fair: Potenziale 603 | 603,
Reisekosten 41

1 68	2 96	3 101	4 114
5 42	6 73	7 78	8 34
9 68	10 51	11 73	12 52
13 51	14 105	15 123	16 77

Kurz: Potenziale 625 | 581,
Reisekosten 24

1 68	2 96	3 101	4 114
5 42	6 73	7 78	8 34
9 68	10 51	11 73	12 52
13 51	14 105	15 123	16 77

Fair achtet nicht auf Wege, kurz nicht auf Fairness (rechts: eine von mehreren).

Der Ausgleich

Jetzt das angekündigte Goal Programming, minimiere die Summe der **relativen** Abweichungen von beiden Idealen:

```
1 modell, X, Y, F1v, F2v = Standortmodell()
2 @objective(modell, Min,
3     (F1stern - F1v)/F1stern + (F2v - F2stern)/F2stern)
4 optimize!(modell)
5
6 println("Standorte: ", [j for j in J if value(Y[j]) > 0.5])
7 println("F1 = ", round{Int, value(F1v)},
8     " (Ideal ", round{Int, F1stern}, "), ",
9     "F2 = ", round{Int, value(F2v)},
10    " (Ideal ", round{Int, F2stern}, ")")
```

Der Ausgleich

Jetzt das angekündigte Goal Programming, minimiere die Summe der **relativen** Abweichungen von beiden Idealen:

```
1 modell, X, Y, F1v, F2v = standortmodell()
2 @objective(modell, Min,
3     (F1stern - F1v)/F1stern + (F2v - F2stern)/F2stern)
4 optimize!(modell)
5
6 println("Standorte: ", [j for j in J if value(Y[j]) > 0.5])
7 println("F1 = ", round{Int, value(F1v)},
8     " (Ideal ", round{Int, F1stern}, "), ",
9     "F2 = ", round{Int, value(F2v)},
10    " (Ideal ", round{Int, F2stern}, ")")
```

Standorte: [8, 9]

F1 = 602 (Ideal 603), F2 = 24 (Ideal 24)

Der Ausgleich

Jetzt das angekündigte Goal Programming, minimiere die Summe der **relativen** Abweichungen von beiden Idealen:

```
1 modell, X, Y, F1v, F2v = Standortmodell()
2 @objective(modell, Min,
3     (F1stern - F1v)/F1stern + (F2v - F2stern)/F2stern)
4 optimize!(modell)
5
6 println("Standorte: ", [j for j in J if value(Y[j]) > 0.5])
7 println("F1 = ", round{Int, value(F1v)},
8     " (Ideal ", round{Int, F1stern}, "), ",
9     "F2 = ", round{Int, value(F2v)},
10    " (Ideal ", round{Int, F2stern}, ")")
```

Standorte: [8, 9]

F1 = 602 (Ideal 603), F2 = 24 (Ideal 24)

Beide Ziele bleiben nahe an ihrem Ideal. Der Preis des Kompromisses ist direkt ablesbar.

Der Kompromiss auf der Karte

Kompromiss: Potenziale 602 | 604,
Reisekosten 24

1 68	2 96	3 101	4 114
5 42	6 73	7 78	8 34
9 68	10 51	11 73	12 52
13 51	14 105	15 123	16 77

- Standorte **8 und 9**.
- $F_1 = 602$ statt 603.
- Reisekosten 24, wie im Ideal.

Der Kompromiss auf der Karte

Kompromiss: Potenziale 602 | 604,
Reisekosten 24

1 68	2 96	3 101	4 114
5 42	6 73	7 78	8 34
9 68	10 51	11 73	12 52
13 51	14 105	15 123	16 77

- Standorte **8 und 9**.
- $F_1 = 602$ statt 603.
- Reisekosten 24, wie im Ideal.

Fast perfekt ausgewogen **und**
kurz: Der Kompromiss kostet hier
fast nichts.

Zusammenfassung

Das Wichtigste auf einen Blick

Das Wichtigste auf einen Blick

- Mehrere Ziele erzeugen oft Zielkonflikte.

Das Wichtigste auf einen Blick

- Mehrere Ziele erzeugen oft Zielkonflikte.
- Lexikographische Ordnung: Ziele strikt nacheinander optimieren.

Das Wichtigste auf einen Blick

- Mehrere Ziele erzeugen oft **Zielkonflikte**.
- **Lexikographische Ordnung**: Ziele strikt nacheinander optimieren.
- **Haupt-/Nebenziele**: eines optimieren, andere als Mindestanforderung.

Das Wichtigste auf einen Blick

- Mehrere Ziele erzeugen oft **Zielkonflikte**.
- **Lexikographische Ordnung**: Ziele strikt nacheinander optimieren.
- **Haupt-/Nebenziele**: eines optimieren, andere als Mindestanforderung.
- **Zielgewichtung** und **Goal Programming** fassen die Ziele über Gewichte bzw. Abstände zusammen.

Das Wichtigste auf einen Blick

- Mehrere Ziele erzeugen oft **Zielkonflikte**.
- **Lexikographische Ordnung**: Ziele strikt nacheinander optimieren.
- **Haupt-/Nebenziele**: eines optimieren, andere als Mindestanforderung.
- **Zielgewichtung** und **Goal Programming** fassen die Ziele über Gewichte bzw. Abstände zusammen.
- Anwendung: **Standortplanung** wägt faire Umsätze gegen Reisekosten ab.

Literatur

Literaturverzeichnis

Domschke, Wolfgang, Andreas Drexl, Robert Klein, und Armin Scholl. 2015. *Einführung in Operations Research*. 9. Aufl. Springer Gabler. <https://doi.org/10.1007/978-3-662-48216-2>.