Vorlesung 10: Mehrfache Zielsetzung

Grundlagen Operations Research

Mehrfache Zielsetzung

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

$$\begin{aligned} \max F &= 20X_1 + 30X_2 \\ X_1 + X_2 &\le 5, \quad X_1 + 3X_2 \le 9, \quad X_1, X_2 \ge 0 \end{aligned}$$

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

Deckungsbeiträge

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

$$\mathrm{DB}_1 = 20 - 1 \cdot 5 = 15, \qquad \mathrm{DB}_2 = 30 - 3 \cdot 5 = 15$$

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

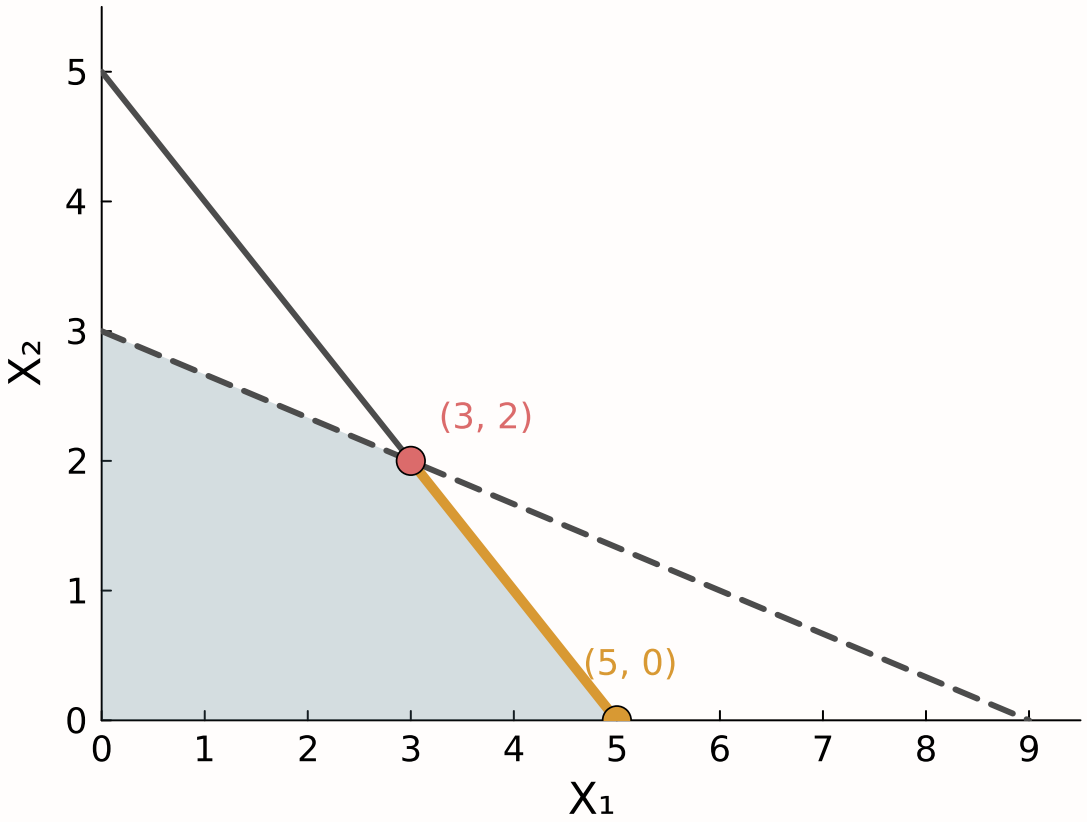
Ziel 2: Deckungsbeitrag

$$\begin{aligned} \max F &= 15X_1 + 15X_2 \\ X_1 + X_2 &\le 5, \quad X_1 + 3X_2 \le 9, \quad X_1, X_2 \ge 0 \end{aligned}$$

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

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: $(5, 0)$ (gelb).

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

Frage: **Wie lösen wir den Konflikt auf?**

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**.

Lexikographische Ordnung

Ziele nach Wichtigkeit

Eine strikte Rangfolge der Ziele $A \succ B \succ C \succ \ldots$

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.

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

$$\begin{aligned} \max F_B &= 20X_1 + 30X_2 \\ 15X_1 + 15X_2 &\geq 75 \quad (\text{DB-Niveau halten}) \\ X_1 + X_2 &\leq 5, \quad X_1 + 3X_2 \leq 9 \end{aligned}$$

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

Haupt- und Nebenziele

Hauptziel plus Nebenziel

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

$$\begin{aligned} \max F &= 15X_1 + 15X_2 && (\text{Hauptziel: DB}) \\ 20X_1 + 30X_2 &\geq 110 && (\text{Nebenziel: Umsatz}) \\ X_1 + X_2 &\leq 5, \quad X_1 + 3X_2 \leq 9 \end{aligned}$$

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

Zielgewichtung

Gewichtete Summe

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

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

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

Beispiel

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

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

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

Goal Programming

Abstand zum Ideal

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

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

- Summe der quadrierten Abstände: $Z_3 = \sum_i \lambda_i (F_i^* - F_i)^2$.

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:

$$Z_1 = (75 - 15X_1 - 15X_2) + (120 - 20X_1 - 30X_2) \to \min$$

$$\Leftrightarrow \quad \max 35X_1 + 45X_2$$

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

Der Min-Max-Ansatz

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

$$\begin{aligned} &\min Z \\ &\quad Z + 20X_1 + 30X_2 \geq 120 \qquad (\text{Abstand zum Umsatzmaximum}) \\ &\quad Z + 15X_1 + 15X_2 \geq 75 \qquad (\text{Abstand zum DB-Maximum}) \\ &\quad X_1 + X_2 \leq 5, \quad X_1 + 3X_2 \leq 9 \end{aligned}$$

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

> ! Ein robuster Kompromiss
>
> **Alle** Ansätze empfehlen hier $(3, 2)$: ein stabiler Ausgleich beider Ziele.

Kontrolle mit Julia

Goal Programming mit JuMP

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

```
using JuMP, HiGHS

# Einzeloptima bestimmen
ideal(c1, c2) = begin
    modell = Model(HiGHS.Optimizer); set_silent(modell)
    @variable(modell, X[1:2] >= 0)
    @objective(modell, Max, c1*X[1] + c2*X[2])
    @constraint(modell, X[1] + X[2] <= 5); @constraint(modell, X[1] + 3X[2]
<= 9)
    optimize!(modell); objective_value(modell)
end
F1, F2 = ideal(20, 30), ideal(15, 15)     # Umsatz-, DB-Maximum
```

```
modell = Model(HiGHS.Optimizer); set_silent(modell)    # minimiere max.
Abweichung
@variable(modell, X[1:2] >= 0); @variable(modell, Z >= 0)
@constraint(modell, Z + 20X[1] + 30X[2] >= F1)
@constraint(modell, Z + 15X[1] + 15X[2] >= F2)
@constraint(modell, X[1] + X[2] <= 5); @constraint(modell, X[1] + 3X[2] <= 9)
@objective(modell, Min, Z)
optimize!(modell)
println("X = ", value.(X), ", Z* = ", objective_value(modell))
```

```
X = [2.999999999999999, 2.000000000000001], Z* = 0.0
```

Anwendung: Standortplanung

Ein Gebietsplanungsproblem

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

68	96°	101	114
42	73	78	34°
68°	51	73	52
51	105	123°	77

Variablen und Restriktionen

Frage: **Welche Variablen und Restriktionen brauchen wir?**

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

> ℹ Fairness gegen Kosten
>
> - F_1 = **kleinstes** zugeordnetes Umsatzpotenzial $\to$ **maximieren** (ausgewogene Gebiete).
> - F_2 = gesamte **Reisekosten** $\sum_{ij} d_{ij} X_{ij} \to$ **minimieren**.

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

Die Einzeloptima

Erst die beiden Ideale, jeweils mit demselben Grundmodell:

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

function standortmodell()
    modell = Model(HiGHS.Optimizer); set_silent(modell)
    @variable(modell, Y[J], Bin); @variable(modell, X[I,J], Bin)
    @variable(modell, F1 >= 0)
    @constraint(modell, sum(Y[j] for j in J) == 2)
    @constraint(modell, [i in I, j in J], X[i,j] <= Y[j])
    @constraint(modell, [j in J], X[j,j] == Y[j])
    @constraint(modell, [i in I], sum(X[i,j] for j in J) == 1)
    @constraint(modell, [j in J],
        F1 <= sum(u[i]*X[i,j] for i in I) + sum(u)*(1 - Y[j]))
    modell, X, Y, F1, sum(d(i,j)*X[i,j] for i in I, j in J)
end

m1, _, _, F1v, _ = standortmodell()
@objective(m1, Max, F1v); optimize!(m1); F1stern = objective_value(m1)
m2, _, _, _, F2v = standortmodell()
@objective(m2, Min, F2v); optimize!(m2); F2stern = objective_value(m2)
println("F1* = ", round(Int, F1stern), ", F2* = ", round(Int, F2stern))
```

```
F1* = 603, F2* = 24
```

Der Ausgleich

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

```
modell, X, Y, F1v, F2v = standortmodell()
@objective(modell, Min,
    (F1stern - F1v)/F1stern + (F2v - F2stern)/F2stern)
optimize!(modell)

println("Standorte: ", [j for j in J if value(Y[j]) > 0.5])
println("F1 = ", round(Int, value(F1v)), " (Ideal ", round(Int, F1stern),
      "), F2 = ", round(Int, value(F2v)), " (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.

Zusammenfassung

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

Bibliografie