

Vorlesung 07: Transportproblem

Grundlagen Operations Research

Minimaler spannender Baum

Ein Netz für sechs Städte

Aufgabe und Definition

Gesucht ist ein **gesamtkostenminimales** Streckennetz, das n Orte direkt oder indirekt verbindet.

Minimaler spannender Baum

In einem zusammenhängenden, bewerteten, ungerichteten Graphen G ist T^* ein **minimaler spannender Baum**, wenn kein anderer spannender Baum eine **geringere** Gesamtbewertung besitzt.

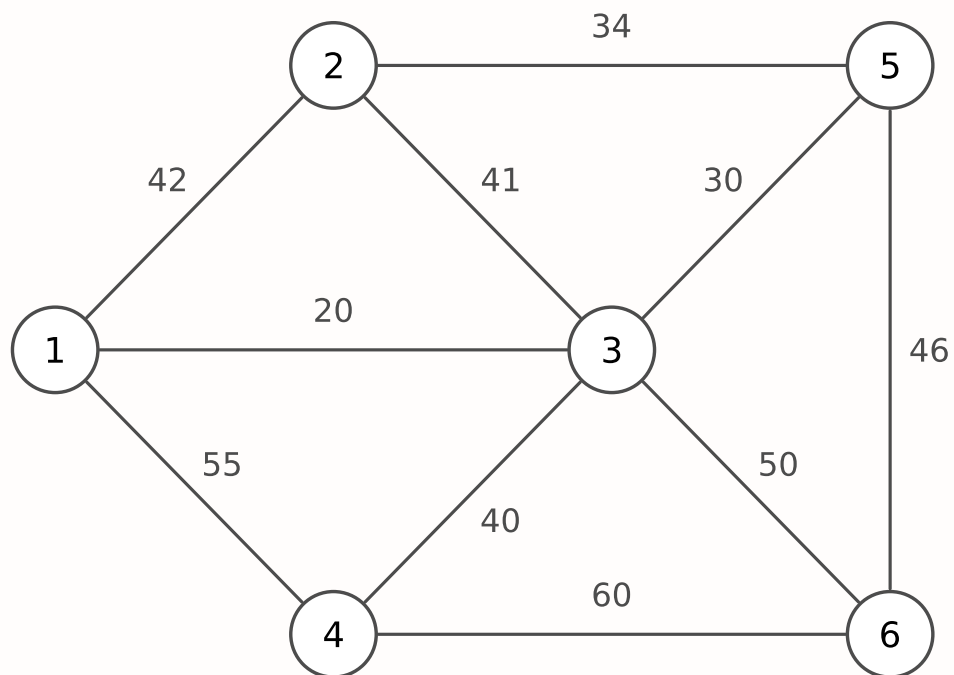
Der Kruskal-Algorithmus

Idee

Kanten **aufsteigend** nach Bewertung durchgehen und jede Kante aufnehmen, die **keinen Kreis** erzeugt, bis $n - 1$ Kanten gewählt sind.

- **Start:** Kanten sortieren, $\lambda(k_1) \leq \lambda(k_2) \leq \dots$
- **Iteration:** k_μ aufnehmen, falls kein Kreis entsteht.
- **Abbruch:** sobald $\bar{\mathcal{E}}$ genau $n - 1$ Kanten enthält.

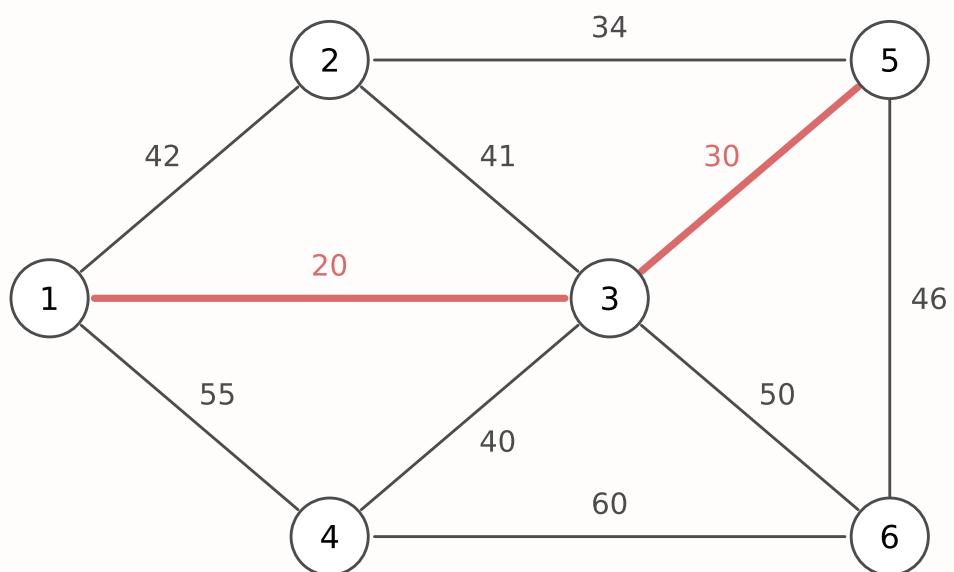
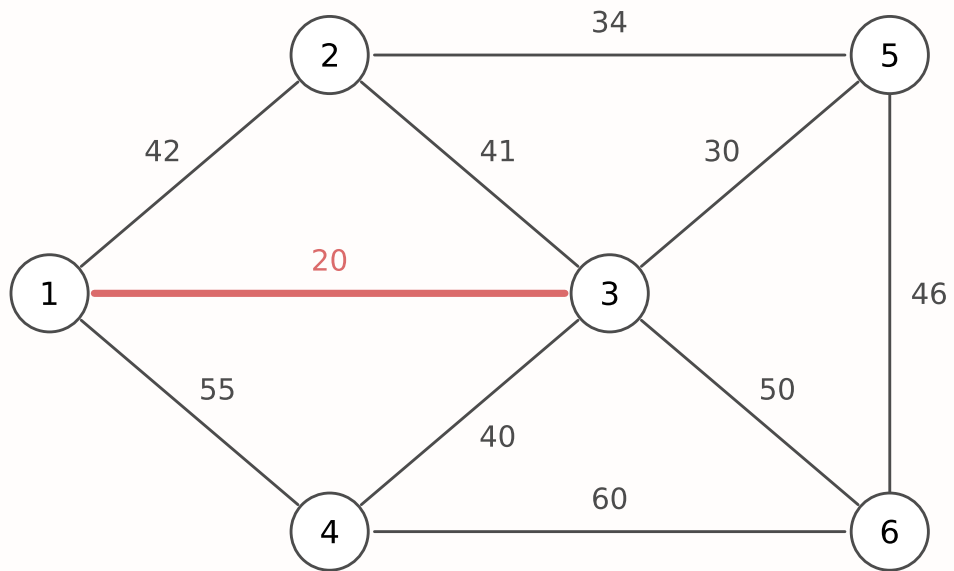
Beispiel: Flugstreckennetz

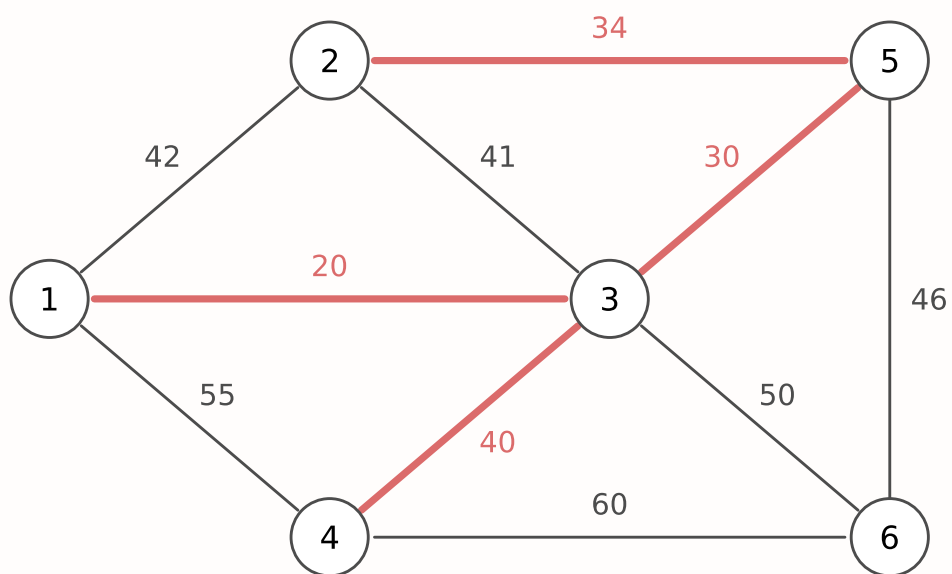
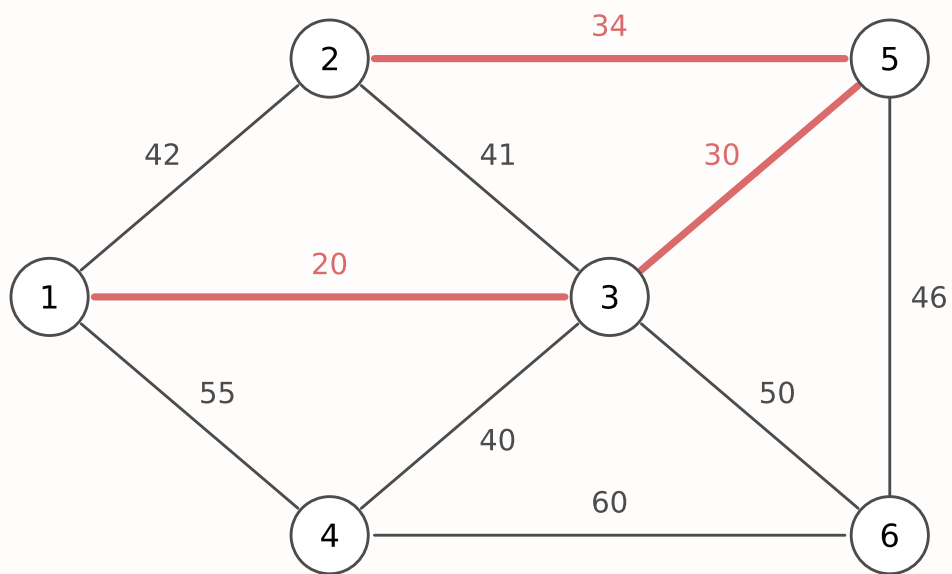


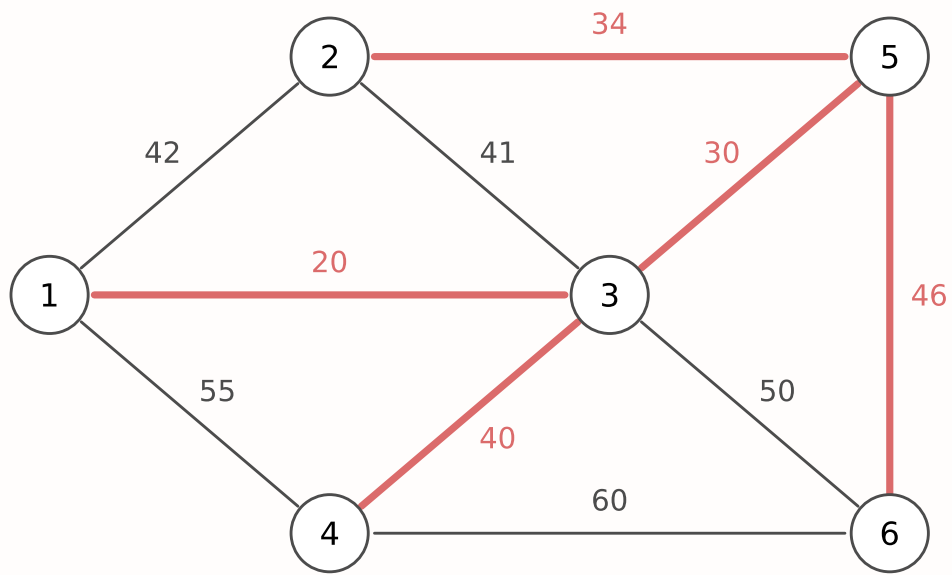
Sechs Städte, zehn mögliche Verbindungen mit Baukosten.

Frage: **Welche fünf Kanten würden Sie wählen?**

Kruskal Schritt für Schritt







Günstigste Kante zuerst: $[1, 3]$ mit 20.

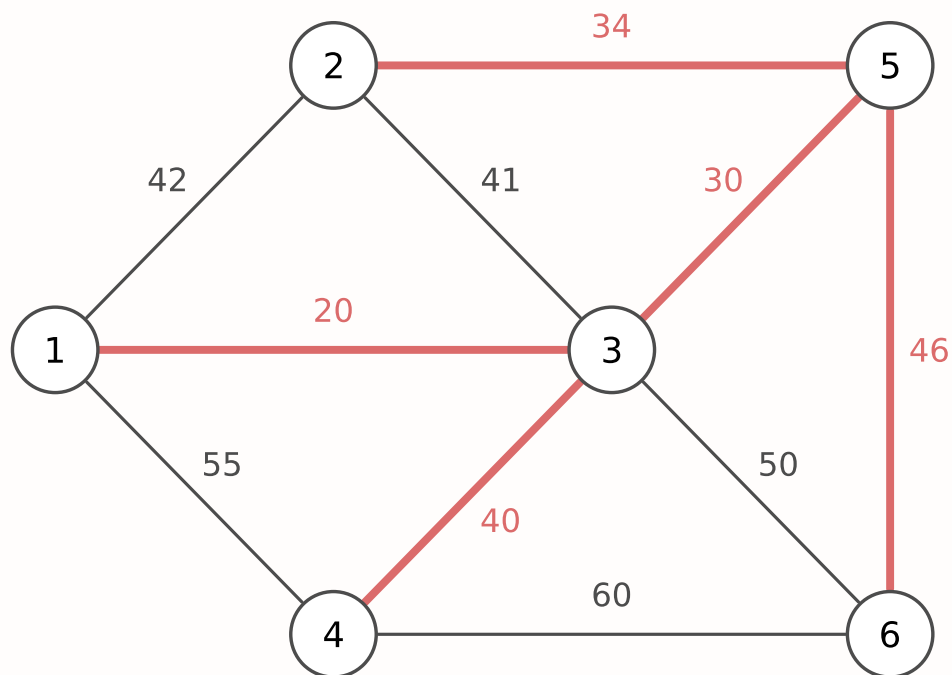
$[3, 5]$ mit 30: kein Kreis, aufnehmen.

$[2, 5]$ mit 34: kein Kreis, aufnehmen.

$[3, 4]$ mit 40: aufnehmen. Danach $[2, 3]$ (41) und $[1, 2]$ (42): beide würden einen **Kreis** schließen, also überspringen!

$[5, 6]$ mit 46, die fünfte Kante: $n - 1 = 5$, **fertig**.

Der minimale spannende Baum



! Ergebnis

Kanten $[1, 3]$, $[3, 5]$, $[2, 5]$, $[3, 4]$, $[5, 6]$ mit Gesamtbewertung

$$20 + 30 + 34 + 40 + 46 = 170$$

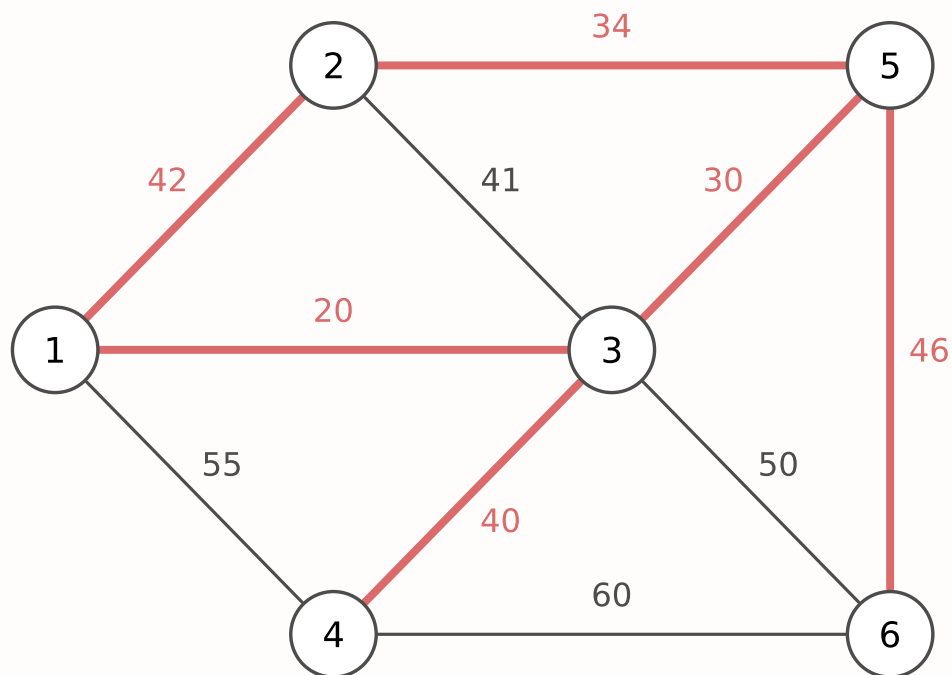
Minimaler 1-Baum

Ein **1-Baum** ist zusammenhängend mit **genau einem** Kreis.

1. Entferne einen vorgegebenen Knoten i_0 (kein Artikulationsknoten) samt seiner Kanten.
2. Bestimme für den Rest mit **Kruskal** einen minimalen spannenden Baum.
3. Füge i_0 mit seinen **zwei günstigsten** Kanten wieder hinzu.

Der 1-Baum liefert eine **untere Schranke** für Rundreiseprobleme (Travelling Salesman).

1-Baum am Beispiel



Für $i_0 = 1$:

- Ohne Knoten 1 wählt Kruskal vier Kanten (Kosten 150).
- Knoten 1 kehrt mit $[1, 3]$ (20) und $[1, 2]$ (42) zurück.

Ein Kreis: $1 - 3 - 5 - 2 - 1$. Gesamt: 212.

Das Transportproblem

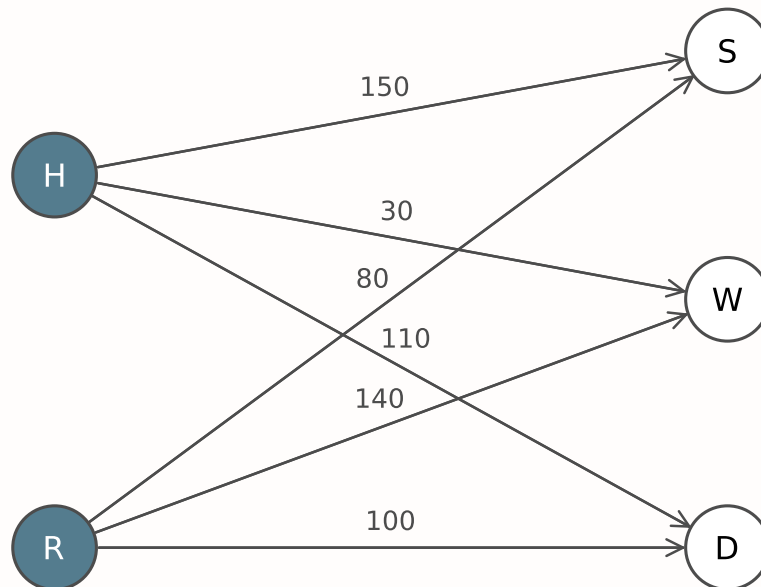
Beispiel: Reifentransport

Ein Unternehmen produziert in **Hannover** und **Regensburg** und beliefert **Stuttgart**, **Wolfsburg** und **Dresden**.

von / nach	Stuttgart	Wolfsburg	Dresden	Angebot
Hannover	150	30	110	15
Regensburg	80	140	100	15
Nachfrage	13	11	6	30

Summe Angebot = Summe Nachfrage = 30.

Als Netz



Anbieter **H, R** (Angebot a_i) links, Nachfrager **S, W, D** (Bedarf b_j) rechts.
Jeder Pfeil trägt die **Transportkosten** c_{ij} pro Ladung.

Lineares Optimierungsproblem

$$\text{Minimiere } F = \sum_i \sum_j c_{ij} X_{ij}$$

$$\sum_j X_{ij} = a_i \text{ für alle } i \quad (u_i)$$

$$\sum_i X_{ij} = b_j \text{ für alle } j \quad (v_j)$$

$$X_{ij} \geq 0$$

X_{ij} : Transportmenge von Standort i zu Abnehmer j .

Eine Gleichung ist redundant

Die Angebots- und Nachfragegleichungen sind **nicht** unabhängig:

$$G_1 = G_3 + G_4 + G_5 - G_2$$

Eine Gleichung darf **entfallen**. Bei $|I| + |J|$ Restriktionen bleiben $|I| + |J| - 1$ unabhängige. Die Dualvariablen sind deshalb nur bis auf eine **Normierung** bestimmt (gleich: $u_1 := 0$).

Angebot ungleich Nachfrage

i Ausgleich

Ist $\sum_i a_i > \sum_j b_j$, führt man einen **fiktiven Nachfrager** ein mit

$$b_{|J|+1} = \sum_i a_i - \sum_j b_j, \quad c_{i, |J|+1} = 0$$

So wird jedes Transportproblem auf den **ausgeglichenen** Fall zurückgeführt.

Vogelsche Approximationsmethode

Idee der Methode

Heuristik für eine **gute Startlösung**:

💡 Idee

Betrachte je Zeile und Spalte die **Strafkosten** Δ : die Differenz aus günstigster und zweitgünstigster Verbindung. Zuerst wird dort belegt, wo Δ am **größten** ist (dort wäre Ausweichen am teuersten). Bei Gleichstand entscheidet der kleinere Index (zuerst Zeilen, dann Spalten).

1. Iteration

von / nach	S	W	D	a_i	Δ_i
Hannover	150	30	110	15	80
Regensburg	80	140	100	15	20
b_j	13	11	6		
Δ_j	70	110	10		

$\max \Delta = 110$ (Spalte W) $\Rightarrow X_{1,2} = \min\{15, 11\} = 11$.

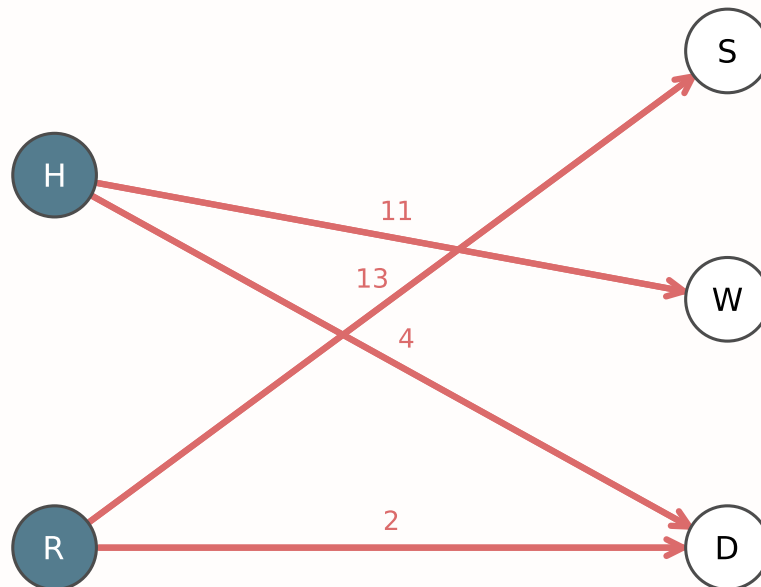
2. und 3. Iteration

- Spalte W ist gedeckt. Größtes $\Delta = 70$ (Spalte S) $\Rightarrow X_{2,1} = \min\{15, 13\} = 13$.
- Es bleibt nur noch Dresden: $X_{1,3} = 4$, $X_{2,3} = 2$.

! Startlösung

$$F = 30 \cdot 11 + 80 \cdot 13 + 110 \cdot 4 + 100 \cdot 2 = 2010$$

Lösung als Baum



- Je **Basisvariable** eine Kante.
- Anzahl Kanten = Knoten -1 (hier $4 = 5 - 1$).
- Eine zusätzliche Kante erzeugt einen **Umverteilungskreis**.

MODI-Methode

Reduzierte Kosten

Spezialisiertes LP-Verfahren auf einer zulässigen Startlösung. Die reduzierten Kosten einer Transportvariablen:

$$\bar{c}_{ij} = c_{ij} - u_i - v_j$$

Für **Basisvariablen** gilt stets $\bar{c}_{ij} = 0$, also $c_{ij} = u_i + v_j$. Daraus bestimmt man die Dualvariablen u_i, v_j .

Dualvariablen bestimmen

Aus den vier Basisvariablen und $u_1 := 0$:

$$30 = u_1 + v_2$$

$$110 = u_1 + v_3$$

$$80 = u_2 + v_1$$

$$100 = u_2 + v_3 \Rightarrow v_2 = 30, v_3 = 110, u_2 = -10, v_1 = 90$$

Optimalitätsprüfung

Reduzierte Kosten der **Nichtbasisvariablen**:

$$\bar{c}_{1,1} = 150 - 0 - 90 = 60 > 0$$

$$\bar{c}_{2,2} = 140 + 10 - 30 = 120 > 0$$

Grafisch ist $\bar{c}_{1,1}$ der **Umweg über den Kreis** $(H, S) \rightarrow (H, D) \rightarrow (R, D) \rightarrow (R, S)$: $150 - 110 + 100 - 80 = 60$.

Alle $\bar{c}_{ij} \geq 0 \Rightarrow$ die **Startlösung ist bereits optimal**. Wäre ein $\bar{c}_{ij} < 0$, würde man X_{ij} entlang des Kreises in die Basis bringen, wie im nächsten Beispiel.

MODI mit Umverteilung

Angebot $\neq$ Nachfrage

Drei Standorte, vier Zentren. Das Angebot **übersteigt** die Nachfrage:

$$\sum a_i = 160 > \sum b_j = 140$$

Ausgleich durch einen **fiktiven Nachfrager** $j = 5$ mit $b_5 = 20$ und Kosten $c_{i5} = 0$:

c_{ij}	1	2	3	4	5	a_i
1	110	150	160	130	0	50
2	130	140	180	120	0	60
3	140	100	110	160	0	50
b_j	30	40	20	50	20	

Die Startlösung

Der Distributionsleiter plant bisher so (Basiszellen mit Mengen, $3 + 5 - 1 = 7$ Stück):

X_{ij}	1	2	3	4	5	a_i
1	30	–	10	–	10	50
2	–	–	–	50	10	60
3	–	40	10	–	–	50
b_j	30	40	20	50	20	

Kosten dieses Plans: $30 \cdot 110 + 10 \cdot 160 + 50 \cdot 120 + 40 \cdot 100 + 10 \cdot 110 = 16\,000$.

Frage: **Geht es günstiger?**

Startlösung nicht optimal

Aus den sieben Basiszellen ($c_{ij} = u_i + v_j$, Normierung $u_1 := 0$) folgen $u = (0, 0, -50)$ und $v = (110, 150, 160, 120, 0)$.

Eine Nichtbasisvariable hat **negative** reduzierte Kosten:

$$\bar{c}_{2,2} = c_{2,2} - u_2 - v_2 = 140 - 0 - 150 = -10 < 0$$

$\Rightarrow X_{2,2}$ in die Basis aufnehmen, entlang eines **Umverteilungskreises** durch die Basiszellen.

Der Umverteilungskreis

X_{ij}	1	2	3	4	5	a_i
1	30	–	10 – Δ	–	10 + Δ	50
2	–	+ Δ	–	50	10 – Δ	60
3	–	40 – Δ	10 + Δ	–	–	50
b_j	30	40	20	50	20	

Jede Zeilen- und Spaltensumme bleibt erhalten. Das größtmögliche Δ bestimmen die **– Δ -Zellen**: $\min\{10, 10, 40\} = 10$.

Ersparnis: $\Delta \cdot |\bar{c}_{2,2}| = 10 \cdot 10 = 100$.

Optimale Lösung

Nach der Umverteilung sind **alle** reduzierten Kosten ≥ 0 :

Zelle	$X_{1,1}$	$X_{2,2}$	$X_{2,4}$	$X_{3,2}$	$X_{3,3}$	$X_{1,5}$	$X_{2,5}$
Menge	30	10	50	30	20	20	0

Zwei – Δ -Zellen wurden gleichzeitig null, nur **eine** verlässt die Basis: $X_{2,5} = 0$ bleibt als **degenerierte** Basisvariable (primale Degeneration, Vorlesung 5!).

! Optimum

$X_{1,5} + X_{2,5} = 20$ bleibt beim fiktiven Nachfrager (nicht ausgeliefert),

$$F^* = 16\,000 - 100 = 15\,900$$

Kontrolle mit Julia

Reifentransport mit JuMP

```
using JuMP, HiGHS

a = [15, 15]                # Angebot: Hannover, Regensburg
b = [13, 11, 6]             # Nachfrage: Stuttgart, Wolfsburg, Dresden
c = [150 30 110; 80 140 100] # Transportkosten c[i,j]

modell = Model(HiGHS.Optimizer); set_silent(modell)
@variable(modell, X[1:2, 1:3] >= 0)
@objective(modell, Min, sum(c[i,j] * X[i,j] for i in 1:2, j in 1:3))
@constraint(modell, [i in 1:2], sum(X[i,j] for j in 1:3) == a[i]) # Angebot
@constraint(modell, [j in 1:3], sum(X[i,j] for i in 1:2) == b[j]) # Nachfrage
```

```
optimize!(modell)
println("F* = ", objective_value(modell))
```

F* = 2010.0

Distribution mit JuMP

Angebot als $\leq$ (Überschuss bleibt liegen), Nachfrage als $=$:

```
a = [50, 60, 50]; b = [30, 40, 20, 50]
c = [110 150 160 130; 130 140 180 120; 140 100 110 160]

modell = Model(HiGHS.Optimizer); set_silent(modell)
@variable(modell, X[1:3, 1:4] >= 0)
@objective(modell, Min, sum(c[i,j]*X[i,j] for i in 1:3, j in 1:4))
@constraint(modell, [i in 1:3], sum(X[i,j] for j in 1:4) <= a[i]) # Angebot
@constraint(modell, [j in 1:4], sum(X[i,j] for i in 1:3) == b[j]) # Nachfrage

optimize!(modell)
println("F* = ", round{Int, objective_value(modell)})
```

F* = 15900

Zusammenfassung

Das Wichtigste auf einen Blick

- Der **Kruskal-Algorithmus** liefert den minimalen spannenden Baum (kreisfreies, kostenminimales Netz).
- Das **Transportproblem** verteilt Angebot auf Nachfrage bei minimalen Kosten: eine Restriktion ist redundant.
- Die **Vogelsche Approximationsmethode** erzeugt schnell eine gute Startlösung.
- Die **MODI-Methode** prüft und verbessert sie über reduzierte Kosten $\bar{c}_{ij} = c_{ij} - u_i - v_j$.

Literatur

Literaturverzeichnis

Bibliografie