

Rucksackproblem

Definition 3.8

Das Optimierungsproblem

$$\max \sum_{j=1}^n p_j x_j$$

unter den Neben- und Vorzeichenbedingungen

$$\sum_{j=1}^n w_j x_j \leq C$$

$$x_j \in \{0, 1\} \quad \text{für } j = 1, \dots, n$$

heißt **Rucksackproblem (knapsack problem, KP)**.

Bemerkung:

p_j = Nutzen von Gegenstand j

w_j = Gewicht von Gegenstand j

C = Kapazität des Rucksacks

$x_j = \begin{cases} 1 & \text{wenn Gegenstand } j \text{ ausgewählt wird} \\ 0 & \text{sonst} \end{cases}$

Voraussetzungen (O.B.d.A.):

- p_j, w_j und C sind natürliche Zahlen,
- $\sum_{j=1}^n w_j > C$,
- $w_j \leq C$ für alle $j = 1, \dots, n$.

Wenn nicht anders angegeben gelte außerdem

$$\frac{p_1}{w_1} \geq \frac{p_2}{w_2} \geq \dots \geq \frac{p_n}{w_n},$$

d.h., die Gegenstände sind **absteigend sortiert nach spezifischem Nutzen**.

Relaxation für KP

Die LP-Relaxation von KP ist das **stetige Rucksackproblem (continuous knapsack problem, CKP)**:

$$\begin{aligned} \max \quad & \sum_{j=1}^n p_j x_j \\ \text{u. d. N.} \quad & \sum_{j=1}^n w_j x_j \leq C \\ & 0 \leq x_j \leq 1, \quad j = 1, \dots, n \end{aligned}$$

CKP kann effizient gelöst werden, in dem der **kritische Gegenstand s** ermittelt wird:

$$s = \min \left\{ j : \sum_{i=1}^j w_i > C \right\}$$

Satz 3.9

Die optimale Lösung $\mathbf{x}$ von CKP ist

$$\begin{aligned} x_j &= 1 && \text{für } j = 1, \dots, s-1, \\ x_j &= 0 && \text{für } j = s+1, \dots, n, \\ x_s &= \frac{C - \sum_{j=1}^{s-1} w_j}{w_s} \end{aligned}$$

Folgerung 3.10

Der optimale Zielfunktionswert $z(CKP)$ lautet:

$$z(CKP) = \sum_{j=1}^{s-1} p_j + p_s \frac{C - \sum_{j=1}^{s-1} w_j}{w_s}$$

Weiterhin ist

$$U = \lfloor z(CKP) \rfloor = \left\lfloor \sum_{j=1}^{s-1} p_j + p_s \frac{C - \sum_{j=1}^{s-1} w_j}{w_s} \right\rfloor$$

eine obere Schranke für den Zielfunktionswert $z(KP)$ des Rucksackproblems.

Greedy-Algorithmus

Algorithmus 3.11 (GreedyKP)

```
z := 0
for j := 1 to n do
    if  $C - w_j \geq 0$  then
         $x_j := 1$ 
         $z := z + p_j$ 
         $C := C - w_j$ 
    else
         $x_j := 0$ 
    end
end
```

Eigenschaften des Greedy-Algorithmus

- Es sei $z(GKP)$ der Zielfunktionswert der Lösung, die durch GreedyKP ermittelt wird. Dann gilt:

$$z(GKP) \leq z(KP) \leq U \leq z(GKP) + p_s$$

D.h., der absolute Fehler von GreedyKP ist $\leq p_s$.

- Man betrachte die Folge der KPs mit

$$n = 2, p_1 = w_1 = 1, p_2 = w_2 = k, C = k.$$

Für $k \rightarrow \infty$ gilt $\frac{z(GKP)}{z(KP)} \rightarrow 0$, d.h. die **relative Güte (worst-case performance ratio)** der berechneten Lösung wird beliebig schlecht.

- Verbesserung: Wir vergleichen $z(GKP)$ mit p_s . Es sei

$$z(GKP') = \max\{z(GKP), p_s\}$$

Dann gilt

$$\frac{z(GKP')}{z(KP)} \geq \frac{1}{2}$$

Begründung: Aus $z(KP) \leq z(GKP) + p_s$ folgt $z(KP) \leq 2z(GKP')$.
Damit ist die relative Güte des verbesserten Greedy-Algorithmus nicht mehr beliebig schlecht.

Branch-and-Bound-Algorithmus von Horowitz und Sahni

Algorithmus 3.12 (Horowitz-Sahni (1974))

Allgemeines Vorgehen: Entscheidungen für $j = 1, \dots, n$ sequentiell treffen.

- **Verzweigungsregel:** $x_j = 1$ bzw. $x_j = 0$. Wenn nur eine Entscheidung möglich ist, wird nicht verzweigt.
- **Obere Schranke:** U , unter Berücksichtigung der getroffenen Entscheidungen. Wird nur unmittelbar vor einer Verzweigung berechnet.
- **untere Schranke:** keine.
- **Selektionsstrategie:** Tiefensuche, Zweig $x_j = 1$ wird zuerst untersucht.

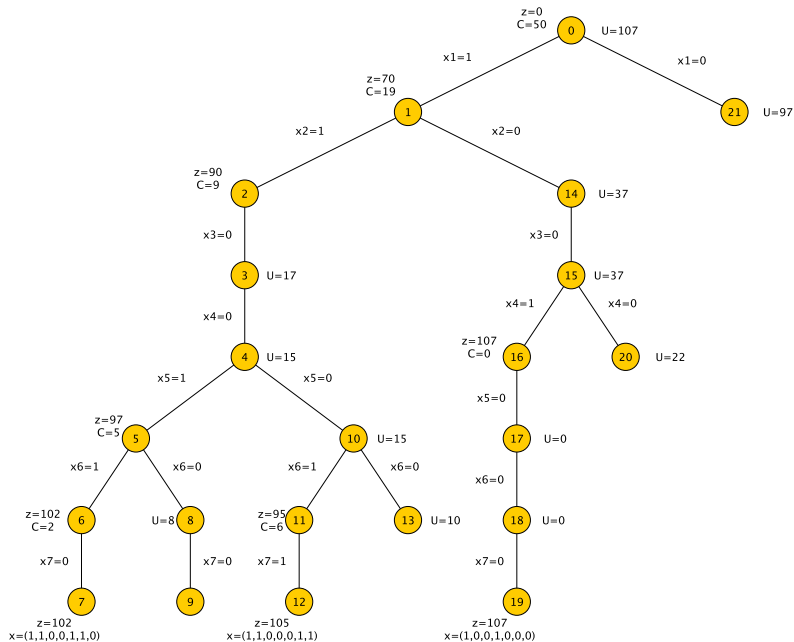
Beispiel zum Algorithmus von Horowitz-Sahni

Beispiel 3.13

$$\begin{aligned}n &= 7, \\(p_j) &= (70, 20, 39, 37, 7, 5, 10), \\(w_j) &= (31, 10, 20, 19, 4, 3, 6), \\C &= 50\end{aligned}$$

In der folgenden Darstellung sind:

- z der aktuelle Zielfunktionswert,
- C die verbleibende Restkapazität,
- U obere Schranke unter Berücksichtigung der bisherigen Entscheidungen,
- $x_j = 0, 1$ Entscheidungen.



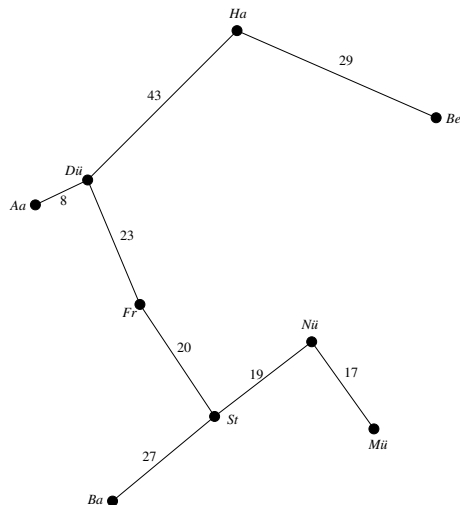
Branch-and-Bound für TSP

Man bestimme eine optimale Lösung für das TSP mit folgender Entfernungsmatrix:

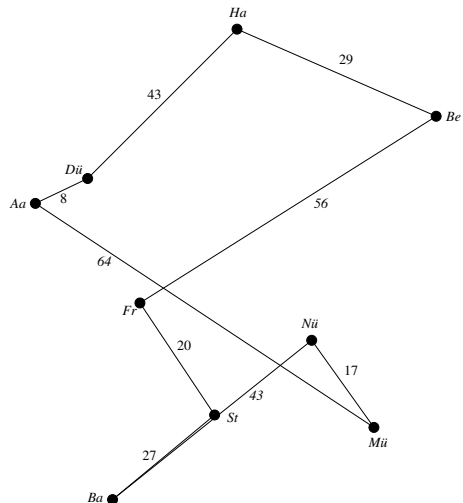
	Aa	Ba	Be	Dü	Fr	Ha	Mü	Nü	St
Aa	0	57	64	8	26	49	64	47	46
Ba	57	0	88	54	34	83	37	43	27
Be	64	88	0	57	56	29	60	44	63
Dü	8	54	57	0	23	43	63	44	41
Fr	26	34	56	23	0	50	40	22	20
Ha	49	83	29	43	50	0	80	63	70
Mü	64	37	60	63	40	80	0	17	22
Nü	47	43	44	44	22	63	17	0	19
St	46	27	63	41	20	70	22	19	0

Obere Schranke: Heuristik

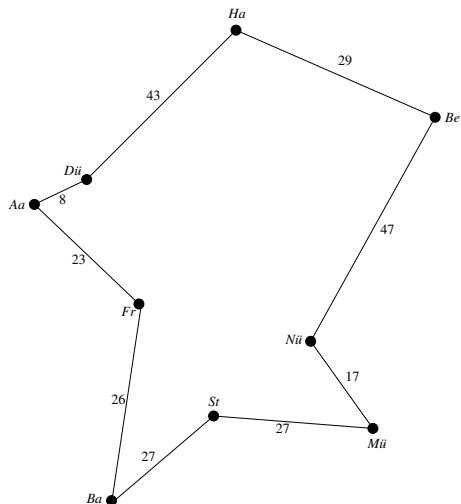
- Berechne **Minimalgerüst (MST)** mit Länge $z(MST) = 186$.
- $z(TSP) \leq 2 * z(MST) = 372$
- Besser: Aus MST eine möglichst gute zulässige Lösung konstruieren.
- Zur Erinnerung: **Tiefensuche auf Minimalgerüst**



- **Tiefensuche** starten in Aachen
- Backtracking
- In der Tour **Abkürzungen bei Backtracking**
- Erlaubt wegen Dreiecksungleichung
- Länge der Tour (erste obere Schranke): **307**



- Weitere Verbesserung durch Kantenaustausch
- 2-opt und 3-opt
- Eliminierung von Kreuzungen
- Länge der Tour (verbesserte obere Schranke): $B_{up} = 250$



Untere Schranke

- Möglichkeit Minimalgerüst, wird hier aber nicht genutzt.
- LP-Relaxation wäre auch möglich, betrachten wir im nächsten Abschnitt.

Bezeichnungen:

- Ab jetzt setzen wir in Entfernungsmatrizen für TSP die Diagonalelemente auf ∞ .
- Für solch eine Entfernungsmatrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ bezeichne $z(TSP_A)$ die Länge einer optimalen TSP-Tour.
- Es sei $\mathbf{A} \in \mathbb{R}^{n \times n}$ eine Entfernungsmatrix für ein TSP sowie $u_1, \dots, u_n \in \mathbb{R}$ und $v_1, \dots, v_n \in \mathbb{R}$ reelle Zahlen. Dann definieren wir die Entfernungsmatrix $\mathbf{B} = (b_{ij})$ durch $b_{ij} = a_{ij} - u_i - v_j$.

Lemma 3.14

$$z(TSP_A) = z(TSP_B) + \sum_{i=1}^n u_i + \sum_{j=1}^n v_j$$

Beweis.

Jede Stadt wird in einer TSP-Tour genau einmal betreten und wieder verlassen. □

Folgerung 3.15

Gilt $\mathbf{B} \geq \mathbf{0}$, dann ist

$$z(TSP_A) \geq \sum_{i=1}^n u_i + \sum_{j=1}^n v_j$$

Beweis.

Aus $\mathbf{B} \geq \mathbf{0}$ folgt $z(TSP_B) \geq 0$ und mit Lemma 3.14

$$z(TSP_A) \geq \sum_{i=1}^n u_i + \sum_{j=1}^n v_j.$$

Damit ist

$$B_{low} = \sum_{i=1}^n u_i + \sum_{j=1}^n v_j$$

eine untere Schranke für $z(TSP_A)$.



Berechnung einer unteren Schranke:

- Ziehe von jeder Zeile i den Maximalwert u_i ab.
- Ziehe anschließend von jeder Spalte j den Maximalwert v_j ab.

$$B_{low} = 8 + 27 + 29 + 8 + 20 + 29 + 17 + 17 + 19 + 8 + 1 = 183$$

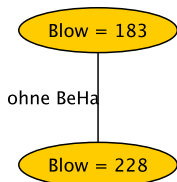
Es entsteht eine neue (jetzt asymmetrische Matrix).

	Aa	Ba	Be	Dü	Fr	Ha	Mü	Nü	St
Aa	∞	41	56	0	17	41	56	39	38
Ba	30	∞	61	27	6	56	10	16	0
Be	35	51	∞	28	26	0	31	15	34
Dü	0	38	49	∞	14	35	55	36	33
Fr	6	6	36	3	∞	30	20	2	0
Ha	20	46	0	14	20	∞	51	34	41
Mü	47	12	43	46	22	63	∞	0	5
Nü	30	18	27	27	4	46	0	∞	2
St	27	0	44	22	0	51	3	0	∞

Verzweigung

- Wir unterteilen eine Lösungsmenge in zwei disjunkte Teilmengen.
- Die erste Teilmenge enthält eine Kante, z.B. Berlin-Hamburg (BeHa).
- Die zweite enthält diese Kante nicht.
- Für die zweite Teilmenge können wir die untere Schranke auf $183 + 30 + 15 = 228$ erhöhen.

Da wir die Kanten als gerichtet betrachten, **genügt es hier (aber nur hier), ausschließlich die Teilmenge mit der höheren unteren Schranke zu durchsuchen.**

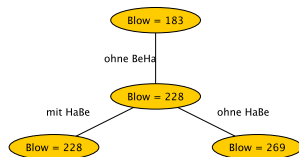


	Aa	Ba	Be	Dü	Fr	Ha	Mü	Nü	St
Aa	∞	41	56	0	17	11	56	39	38
Ba	30	∞	61	27	6	26	10	16	0
Be	20	36	∞	13	11	∞	16	0	19
Dü	0	38	49	∞	14	5	55	36	33
Fr	6	6	36	3	∞	0	20	2	0
Ha	20	46	0	14	20	∞	51	34	41
Mü	47	12	43	46	22	33	∞	0	5
Nü	30	18	27	27	4	16	0	∞	2
St	27	0	44	22	0	21	3	0	∞

Wir unterteilen nun nach HaBe. Für die Lösungsmenge ohne HaBe ergibt sich $B_{low} = 228 + 14 + 27 = 269 \geq B_{up} = 250$.

⇒ Optimale Lösung muss HaBe enthalten.

⇒ Zeile Ha und Spalte Be kann aus der Matrix entfernt werden.

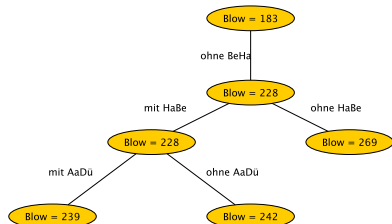


	Aa	Ba	Dü	Fr	Ha	Mü	Nü	St
Aa	∞	41	0	17	11	56	39	38
Ba	30	∞	27	6	26	10	16	0
Be	20	36	13	11	∞	16	0	19
Dü	0	38	∞	14	5	55	36	33
Fr	6	6	3	∞	0	20	2	0
Mü	47	12	46	22	33	∞	0	5
Nü	30	18	27	4	16	0	∞	2
St	27	0	22	0	21	3	0	∞

Als nächstes bietet sich eine Unterteilung auf der Basis von AaDü an.

Ohne AaDü: $B_{low} = 228 + 11 + 3 = 242$

Mit AaDü bedeutet ohne DüAa. Dadurch kann LB für mit AaDü erhöht werden: $B_{low} = 228 + 5 + 6 = 239$

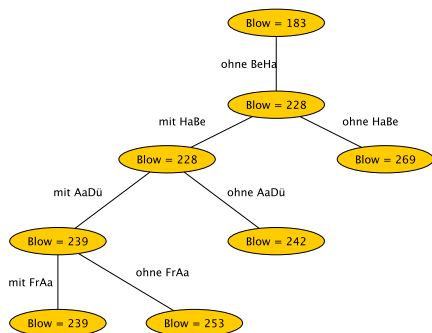


	Aa	Ba	Fr	Ha	Mü	Nü	St
Ba	24	∞	6	26	10	16	0
Be	14	36	11	∞	16	0	19
Dü	∞	33	9	0	50	31	28
Fr	0	6	∞	0	20	2	0
Mü	41	12	22	33	∞	0	5
Nü	24	18	4	16	0	∞	2
St	21	0	0	21	3	0	∞

Als nächstes betrachten wir die Kante FrAa.

Ohne FrAa: $B_{low} = 239 + 14 = 253 \geq 250$. Ausgelotet!

Also mit FrAa, $\Rightarrow$ ohne DüFr.

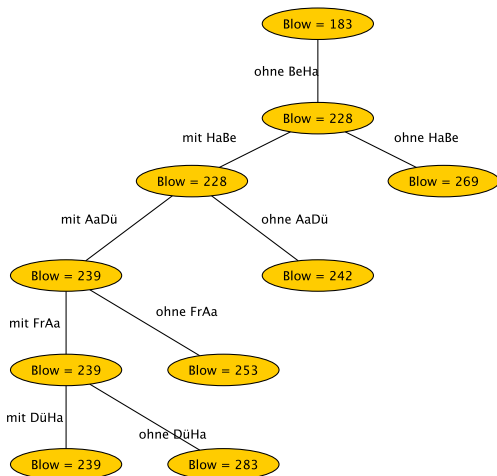


	Ba	Fr	Ha	Mü	Nü	St
Ba	∞	6	26	10	16	0
Be	36	11	∞	16	0	19
Dü	33	∞	0	50	31	28
Mü	12	22	33	∞	0	5
Nü	18	4	16	0	∞	2
St	0	0	21	3	0	∞

Nun die Kante DüHa.

Ohne DüHa: $B_{low} = 239 + 28 + 16 = 283 \geq 250$. Ausgelotet!

Also mit DüHa. Wegen Weg (Fr,Aa,Dü,Ha,Be) folgt ohne BeFr.



	Ba	Fr	Mü	Nü	St
Ba	∞	6	10	16	0
Be	36	∞	16	0	19
Mü	12	22	∞	0	5
Nü	18	4	0	∞	2
St	0	0	3	0	∞

Nun die Kante BeNü.

Ohne BeNü:

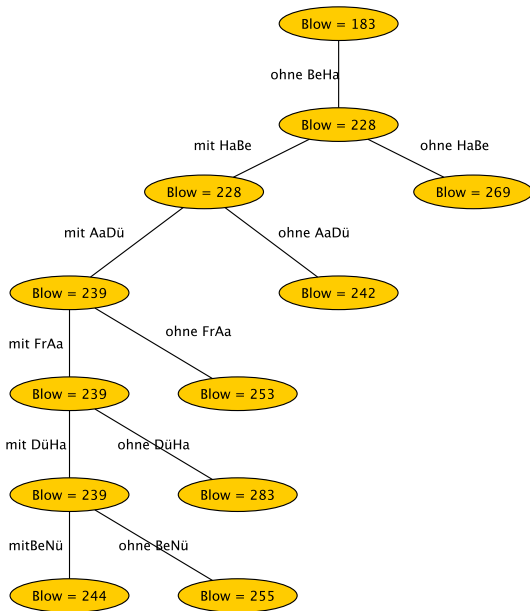
$$B_{low} = 239 + 16 = 255 \geq 250.$$

Ausgelotet!

Mit BeNü:

$$B_{low} = 239 + 5 = 244.$$

$\Rightarrow$ ohne NüFr.



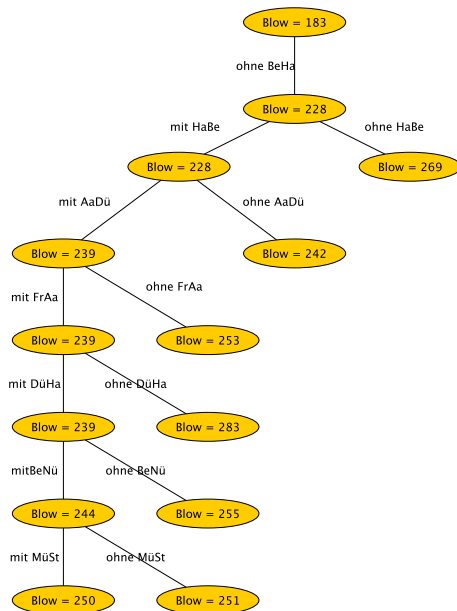
	Ba	Fr	Mü	St
Ba	∞	6	10	0
Mü	7	17	∞	0
Nü	18	∞	0	2
St	0	0	3	∞

Ohne MüSt: $B_{low} = 244 + 7 = 251 \geq 250$.

Ausgelotet!

Mit MüSt:

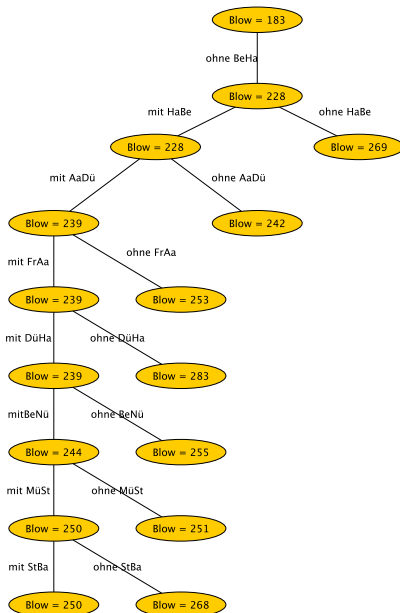
$B_{low} = 244 + 6 = 250$.



	Ba	Fr	Mü
Ba	∞	0	4
Nü	18	∞	0
St	0	0	3

Ohne StBa:

$$B_{low} = 250 + 18 = 269 \geq 250. \text{ Ausgelotet!}$$

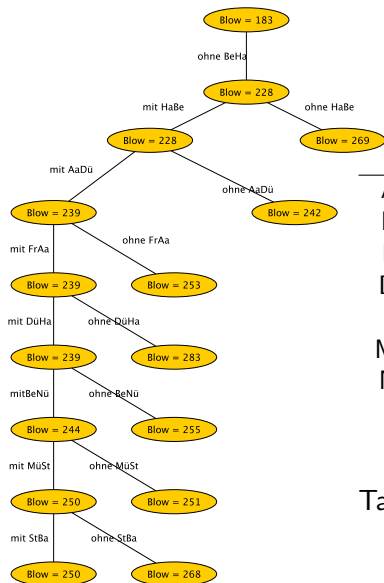


	Fr	Mü
Ba	0	4
Nü	∞	0

Wähle BaFr und NüMü.

$\Rightarrow$ bekannte Tour mit $B_{up} = 250$.

Wir müssen noch den Zweig “Mit HaBe, ohne AaDü” untersuchens mit $B_{low} = 242$.



	Aa	Ba	Dü	Fr	Ha	Mü	Nü	St
Aa	∞	30	∞	6	0	45	28	27
Ba	30	∞	24	6	26	10	16	0
Be	20	36	10	11	∞	16	0	19
Dü	0	38	∞	14	5	55	36	33
Fr	6	6	0	∞	0	20	2	0
Mü	47	12	43	22	33	∞	0	5
Nü	30	18	24	4	16	0	∞	2
St	27	0	19	0	21	3	0	∞

Tafel 

