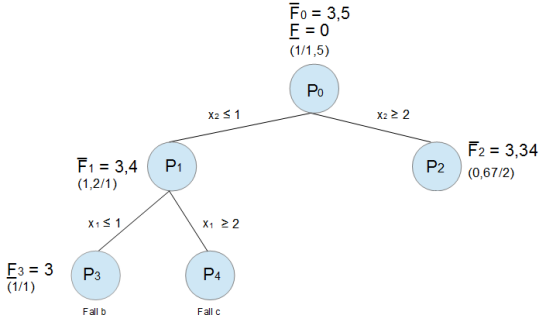
Kapitel 4

Branch-and-Bound



Inhalt

Schranken (1)

Definition 4.1

Gegeben sei ein Maximierungsproblem der Art

$$\max F(\mathbf{x}), \quad \text{u.d.N. } \mathbf{x} \in \mathcal{X}$$

Gilt

$$F(\mathbf{x}) \leq B_{up} \text{ für alle } \mathbf{x} \in \mathcal{X}$$

dann ist B_{up} eine obere Schranke für den optimalen Zielfunktionswert.

Gilt

$$F(\mathbf{x}) \geq B_{low} \text{ für ein } \mathbf{x} \in \mathcal{X}$$

dann ist B_{low} eine untere Schranke für den optimalen Zielfunktionswert.

Schranken (2)

Definition 4.2

Gegeben sein ein Minimierungsproblem der Art

$$\min F(\mathbf{x}), \quad \text{u.d.N. } \mathbf{x} \in \mathcal{X}$$

Gilt

$$F(\mathbf{x}) \geq B_{low} \text{ für alle } \mathbf{x} \in \mathcal{X}$$

dann ist B_{low} eine untere Schranke für den optimalen Zielfunktionswert.

Gilt

$$F(\mathbf{x}) \leq B_{up} \text{ für ein } \mathbf{x} \in \mathcal{X}$$

dann ist B_{up} eine obere Schranke für den optimalen Zielfunktionswert.

Herleitung von Schranken

Für Maximierungsprobleme:

- obere Schranken
  typischerweise durch Relaxationen, z. B. LP-Relaxation bei einem ILP
- untere Schranken
  durch zulässige i. d. R. aber nicht optimale Lösungen, z. B. auf der Basis von Heuristiken

Für Minimierungsprobleme: genau umgekehrt

Wenn nicht anders erwähnt betrachten wir im Folgenden stets Maximierungsprobleme.

Grundprinzip von Branch-and-Bound

Suchverfahren, das die Menge $\mathcal{X}$ der zulässigen Lösungen systematisch durchsucht

Wesentliche Operationen bei der Suche:

- Verzweigung (Branch)

  Teile das Ausgangsproblem P_0 (bzw. die Menge $\mathcal{X}(P_0)$ der zulässigen Lösungen) in zwei Teilprobleme P_1 und P_2 (bzw. Teilmengen $\mathcal{X}(P_1)$ und $\mathcal{X}(P_2)$) auf.
- Beschränkung (Bound)

  Berechne für die Teilprobleme P_1 und P_2 obere Schranken (upper bound) B^1_{up} und B^2_{up} für die optimale Lösung in $\mathcal{X}(P_1)$ und in $\mathcal{X}(P_2)$.

Nutzung von Schranken für die Suche (1)

Es sei B_{low} eine bekannte untere Schranke für $\mathcal{X}(P_0)$, z. B. der Zielfunktionswert einer bekannten zulässigen Lösung $\mathbf{x}$.

Dann gilt:

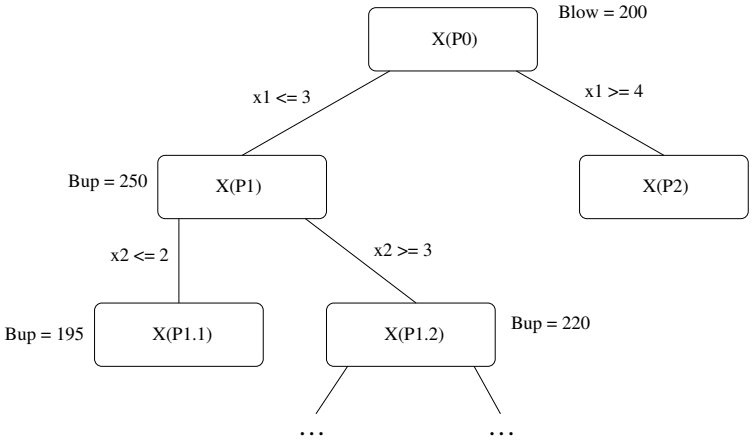
- Aus $B^1_{up} \leq B_{low}$ folgt, dass $\mathcal{X}(P_1)$ keine bessere Lösung als $\mathbf{x}$ enthalten kann.
- Analog für $B^2_{up} \leq B_{low}$

☞ Im Fall von $B^1_{up} \leq B_{low}$ bzw. $B^2_{up} \leq B_{low}$ müssen wir in $\mathcal{X}(P_1)$ bzw. $\mathcal{X}(P_2)$ nicht mehr nach einer optimalen Lösung suchen.

Nutzung von Schranken für die Suche (2)

- Es sei B_{up} eine bekannte obere Schranke für $\mathcal{X}$.
- Findet man ein $\mathbf{x} \in \mathcal{X}$ mit $F(\mathbf{x}) = B_{up}$, dann ist $\mathbf{x}$ eine optimale Lösung.

Branch-and-Bound: Prinzipieller Ablauf für ein ILP



Beispiele: Branch-and-Bound für ILP

Beispiel 4.3

Wir wollen das ILP von Beispiel 3.2 lösen. Es sei dies das Problem P_0.

Die LP-Relaxation hat die optimale Lösung $(\frac{9}{4}, \frac{5}{2})$ mit Zielfunktionswert $\frac{29}{4}$. Dies liefert uns eine allgemeine obere Schranke $B_{up} = \lfloor \frac{29}{4} \rfloor = 7$.

Wir unterteilen das Problem P_0 in zwei disjunkte Teilprobleme:

- Für P_1 gelte die zusätzliche Bedingung $x_1 \geq 3$.
- Für P_2 gelte die zusätzliche Bedingung $x_1 \leq 2$.

Für die LP-Relaxation von P_1 erhalten wir die optimale Lösung $\mathbf{x}^1 = (3, 2)$ mit Zielfunktionswert 7.

- Damit haben wir eine zulässige Lösung gefunden, deren Zielfunktionswert mit $B_{up} = 7$ übereinstimmt.
- Somit ist $\mathbf{x}^1$ eine optimale Lösung für P_0.

Fortsetzung Beispiel.

Wir brauchen P_2 nicht mehr zu lösen. Wenn wir trotzdem die zugehörige LP-Relaxation lösen,

- können wir $B_{low} = 7$ setzen, da $\mathbf{x}^1$ eine zulässige Lösung ist und
- wir erhalten $\mathbf{x}^2 = (2, \frac{5}{2})$ mit Zielfunktionswert $B^2_{up} = 7 \leq B_{low}$ als optimale Lösung für die LP-Relaxation von P_2.
- Hieraus folgt, dass in $\mathcal{X}(P_2)$ keine bessere Lösung als $\mathbf{x}^1$ enthalten sein kann.

Beispiel 4.4

Wir betrachten das ILP

$$\max x_1 + x_2$$

unter den Neben- und Vorzeichenbedingungen

$$\begin{array}{rcrcl} 4x_1 & + & x_2 & \leq & 20 \\ & & 4x_2 & \leq & 10 \\ 2x_1 & + & 3x_2 & \leq & 12 \\ & & x_1, x_2 & \geq & 0 \\ & & x_1, x_2 & \in & \mathbb{Z} \end{array}$$

Tafel ✎

Beispiel 4.5 (Winston, Kapitel 9.3)

Die Telfa Corporation produziert Tische und Stühle.

Für die Produktion eines Tisches werden eine Arbeitsstunde und 9 Quadratmeter Holz benötigt, ein Stuhl erfordert eine Arbeitsstunde und 5 Quadratmeter Holz. Es stehen 6 Arbeitsstunden und 45 Quadratmeter Holz zur Verfügung.

Der Gewinn für einen Tisch beträgt 8 €, für einen Stuhl 5 €.

Tafel ✎

Beispiel 4.6 (Nickel et al., Kapitel 5.4.1)

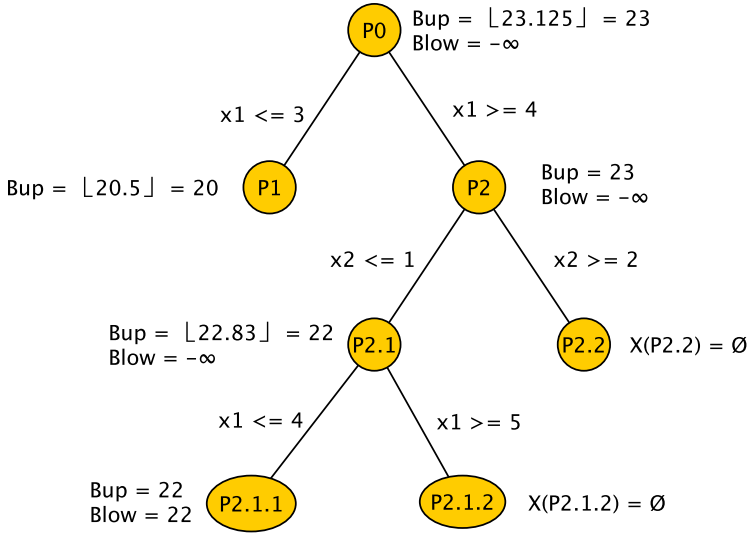
Gegeben sei das ILP

$$\max 5x_1 + 2x_2$$

unter den Neben- und Vorzeichenbedingungen

$$\begin{array}{rcrcl} 6x_1 & + & 2x_2 & \leq & 27 \\ 4x_1 & + & 4x_2 & \leq & 23 \\ & & x_1, x_2 & \geq & 0 \\ & & x_1, x_2 & \in & \mathbb{Z} \end{array}$$

Tafel: Mit Selektionsstrategie “Maximum Upper Bound” ✎


P0
Bup = ⌊23.125⌋ = 23
Blow = −∞
x1 <= 3
x1 >= 4
Bup = ⌊20.5⌋ = 20
P1
P2
Bup = 23
Blow = −∞
x2 <= 1
x2 >= 2
Bup = ⌊22.83⌋ = 22
Blow = −∞
P2.1
P2.2
X(P2.2) = Ø
x1 <= 4
x1 >= 5
Bup = 22
Blow = 22
P2.1.1
P2.1.2
X(P2.1.2) = Ø

Was brauchen wir für Branch-and-Bound?

- Verzweigungsregel: Wie teilen wir die Menge der zulässigen Lösungen auf?
- Obere Schranke: Effizientes Verfahren zur Berechnung einer oberen Schranke für ein Teilproblem, z. B. eine geeignete LP-Relaxation.
- Optionale untere Schranke: Eine Heuristik zur Ermittlung einer guten zulässigen Lösung für ein Teilproblem (üblicherweise auf Basis der optimalen Lösung des relaxierten Problems).
- Selektionsstrategie: In welcher Reihenfolge werden Teilprobleme abgearbeitet? Tiefensuche? Breitensuche? Gesteuert durch die oberen Schranken?

Verzweigung (Branch)

- Ein Problem P_0 wird in k Teilprobleme $P_1, \ldots, P_k$ unterteilt, so dass für die Mengen $\mathcal{X}(P_i)$ der zulässigen Lösungen gilt:

$$\mathcal{X}(P_0) = \bigcup_{i=1}^{k} \mathcal{X}(P_i)$$

und

$$\mathcal{X}(P_i) \cap \mathcal{X}(P_j) = \emptyset \text{ für } i \neq j$$

- Die Probleme $P_1, \ldots, P_k$ werden falls notwendig weiter unterteilt. Dadurch entsteht ein Baum von Problemen mit P_0 als Wurzel.

Übliche Verzweigungsstrategien

Es sei **x** die optimale Lösung der LP-Relaxation von P_0 und die Komponente $x_i = \alpha$ sei nicht ganzzahlig.

- ILP:
  - P_1 erhält die zusätzliche Nebenbedingung $x_i \leq \lfloor \alpha \rfloor$
  - P_2 erhält die zusätzliche Nebenbedingung $x_i \geq \lceil \alpha \rceil$
- kombinatorische Probleme:
  Setze $x_i = 0$ für P_1 bzw. $x_i = 1$ für P_2.

Welches x_i auswählen? Z.B. das am wenigsten bestimmte.

Obere Schranken

- Für jedes Teilproblem P_i bestimmen wir eine obere Schranke (upper bound) B^i_{up}.
- Hierfür lösen wir eine Relaxation P_i^{relax} von P_i, also ein gegenüber P_i vereinfachtes (relaxiertes) Problem (weniger Nebenbedingungen).
- Es muss gelten
$$\mathcal{X}(P_i) \subseteq \mathcal{X}(P_i^{relax})$$
- Wichtig ist, dass die Relaxationen effizient gelöst werden können.
- Für gewöhnliche ILPs benutzt man üblicherweise LP-Relaxationen.
- Für spezielle kombinatorische Probleme können die Relaxationen vereinfachte und effizient lösbare Probleme sein.

  Beispiel: Minimalgerüst ist Relaxation für kürzesten Hamiltonschen Weg

Untere Schranken

- Durch die heuristische Bestimmung einer zulässigen Lösung erhält man eine untere Schranke B_{low}.
- Prinzipiell nicht notwendig. Schlimmstenfalls starten wir mit $B_{low} = -\infty$.
- Ist die optimale Lösung einer Relaxation auch zulässig für das eigentliche Problem P_0, stellt der zugehörige Zielfunktionswert eine untere Schranke dar.

  Beispiel: Bei einem ILP ist die optimale Lösung der LP-Relaxation ganzzahlig.
- B_{low} ist dann im Laufe des Verfahrens gleich dem Zielfunktionswert der besten bekannten zulässigen Lösung von P_0.

Auslotung eines Problems

Definition 4.7

Ein Problem P_i heißt ausgelotet, wenn einer der folgenden Fälle auftritt:

(a) $B^i_{up} \leq B_{low}$

In $\mathcal{X}(P_i)$ kann es keine bessere Lösung geben als die beste bisher bekannte.

(b) $B^i_{up} > B_{low}$ und die optimale Lösung von P_i^{relax} ist zulässig für P_i.

Dann hat man eine neue bisher beste zulässige Lösung für P_0 gefunden. Man setzt nun $B_{low} := B^i_{up}$.

(c) $\mathcal{X}(P_i^{relax}) = \emptyset$

Dann hat P_i^{relax} und damit auch P_i keine zulässige Lösung.

Selektionsstrategie

- Tiefensuche: Durchsuche für ein noch offenes Teilproblem (Knoten im Suchbaum) zuerst den "linken" Teilbaum, dann den "rechten" Teilbaum.

  Vorteile:
  - man erhält i. d. R. schnell eine zulässige Lösung (und damit eine untere Schranke)
  - geringer Speicherplatzverbrauch (kleine Agenda)

  Nachteile:
  - i. d. R. größerer Suchbaum
- Maximum Upper Bound: Untersuche als nächstes das noch offene Teilproblem P_i mit der größten oberen Schranke B^i_{up}.

  Vorteil:
  - Suchbaum i. d. R. kleiner als bei der Tiefensuche

  Nachteil:
  - größerer Speicherplatzverbrauch für Agenda

Rucksackproblem

Definition 4.8

Das Optimierungsproblem

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

unter den Neben- und Vorzeichenbedingungen

$$\begin{aligned} \sum_{j=1}^{n} w_j x_j &\leq C \\ x_j &\in \{0,1\} \quad \text{für } j = 1, \ldots, n \end{aligned}$$

heißt Rucksackproblem (knapsack problem, KP).

Bemerkung:

$$
\begin{aligned}
p_j &= \text{Nutzen von Gegenstand } j \\
w_j &= \text{Gewicht von Gegenstand } j \\
C &= \text{Kapazität des Rucksacks} \\
x_j &= \begin{cases} 1 & \text{wenn Gegenstand } j \text{ ausgewählt wird} \\ 0 & \text{sonst} \end{cases}
\end{aligned}
$$

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 $j = 1, \ldots, n$.

Wenn nicht anders angegeben gelte außerdem

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

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

Relaxation für KP

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

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

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

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

Satz 4.9

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

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

Folgerung 4.10

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 4.11 (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 $k \to \infty$ gilt $\frac{z(GKP)}{z(KP)} \to 0$, 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 4.12 (Horowitz-Sahni (1974))

Allgemeines Vorgehen: Entscheidungen für $j = 1, \ldots, n$ 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.

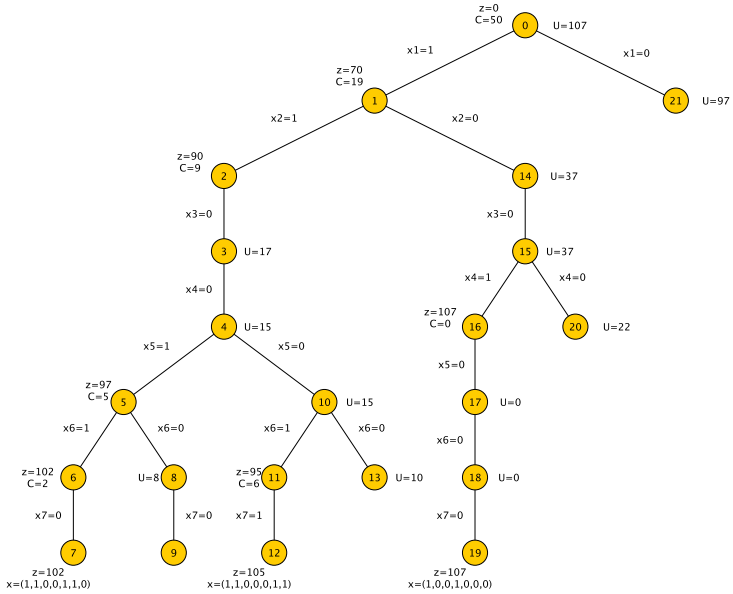
Beispiel zum Algorithmus von Horowitz-Sahni

Beispiel 4.13

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

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.


z=0
C=50
0
U=107
x1=1
x1=0
z=70
C=19
1
21
U=97
x2=1
x2=0
z=90
C=9
2
14
U=37
x3=0
x3=0
3
U=17
15
U=37
x4=0
x4=1
x4=0
z=107
C=0
4
U=15
16
20
U=22
x5=1
x5=0
x5=0
z=97
C=5
5
10
U=15
17
U=0
x6=1
x6=0
x6=1
x6=0
x6=0
z=102
C=2
6
U=8
8
z=95
C=6
11
13
U=10
18
U=0
x7=0
x7=0
x7=1
x7=0
7
9
12
19
z=102
x=(1,1,0,0,1,1,0)
z=105
x=(1,1,0,0,0,1,1)
z=107
x=(1,0,0,1,0,0,0)

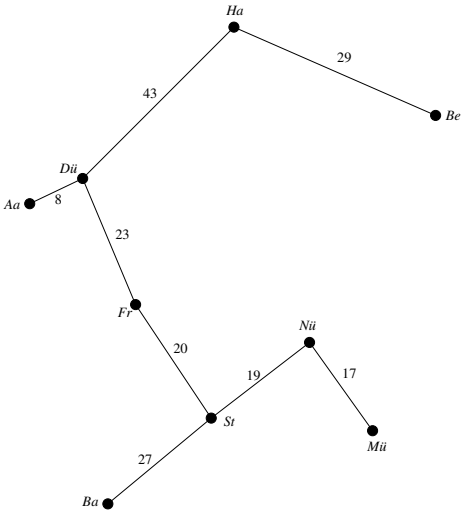
Branch-and-Bound für TSP

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

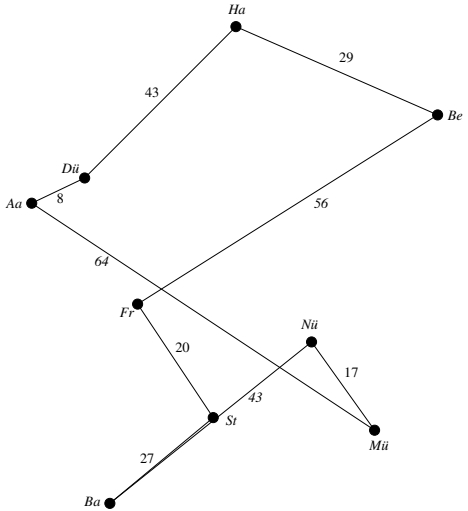
	Aa	Ba	Be	Dü	Fr	Ha	Mü	Nü	St
Aa	0	57	64	8	26	49	64	47	46
Ba	57	0	88	54	34	83	37	43	27
Be	64	88	0	57	56	29	60	44	63
Dü	8	54	57	0	23	43	63	44	41
Fr	26	34	56	23	0	50	40	22	20
Ha	49	83	29	43	50	0	80	63	70
Mü	64	37	60	63	40	80	0	17	22
Nü	47	43	44	44	22	63	17	0	19
St	46	27	63	41	20	70	22	19	0

Obere Schranke: Heuristik

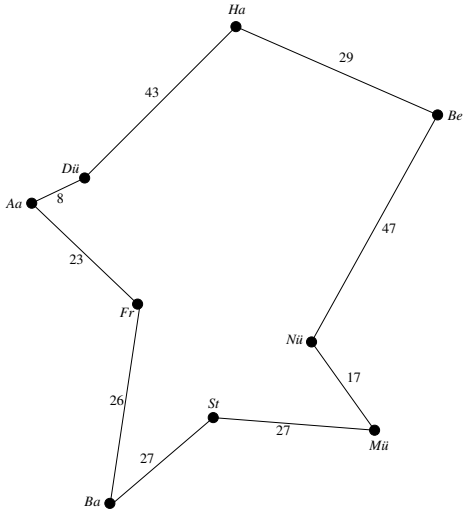
- Berechne Minimalgerüst (MST) mit Länge $z(MST) = 186$.
- $z(TSP) \leq 2 * z(MST) = 372$
- Besser: Aus MST eine möglichst gute zulässige Lösung konstruieren.
- Zur Erinnerung: Tiefensuche auf Minimalgerüst



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



- Weitere Verbesseung 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.

Bezeichungen:

- 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 $u_1, \ldots, u_n \in \mathbb{R}$ und $v_1, \ldots, v_n \in \mathbb{R}$ reelle Zahlen. Dann definieren wir die Entfernungsmatrix $\mathbf{B} = (b_{ij})$ durch $b_{ij} = a_{ij} - u_i - v_j$.

Lemma 4.14

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

Beweis.

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

Folgerung 4.15

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 4.14

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

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.

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

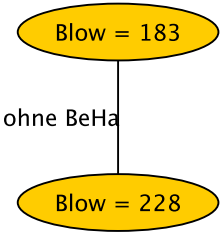
Es entsteht eine neue (jetzt asymmetrische Matrix).

	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.

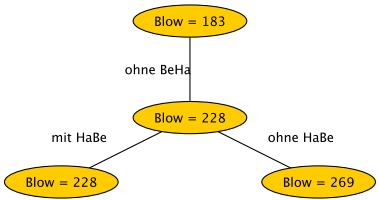


	Aa	Ba	Be	Dü	Fr	Ha	Mü	Nü	St
Aa	∞	41	56	0	17	11	56	39	38
Ba	30	∞	61	27	6	26	10	16	0
Be	20	36	∞	13	11	∞	16	0	19
Dü	0	38	49	∞	14	5	55	36	33
Fr	6	6	36	3	∞	0	20	2	0
Ha	20	46	0	14	20	∞	51	34	41
Mü	47	12	43	46	22	33	∞	0	5
Nü	30	18	27	27	4	16	0	∞	2
St	27	0	44	22	0	21	3	0	∞

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

$\Rightarrow$ Optimale Lösung muss HaBe enthalten.

$\Rightarrow$ Zeile Ha und Spalte Be kann aus der Matrix entfernt werden.

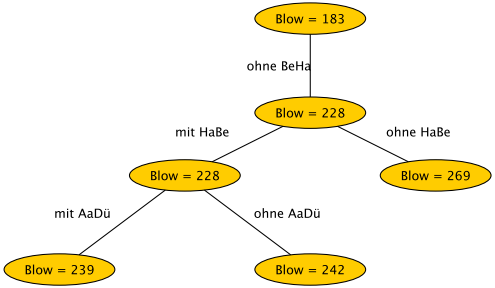


	Aa	Ba	Dü	Fr	Ha	Mü	Nü	St
Aa	∞	41	0	17	11	56	39	38
Ba	30	∞	27	6	26	10	16	0
Be	20	36	13	11	∞	16	0	19
Dü	0	38	∞	14	5	55	36	33
Fr	6	6	3	∞	0	20	2	0
Mü	47	12	46	22	33	∞	0	5
Nü	30	18	27	4	16	0	∞	2
St	27	0	22	0	21	3	0	∞

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:
$B_{low} = 228 + 5 + 6 = 239$

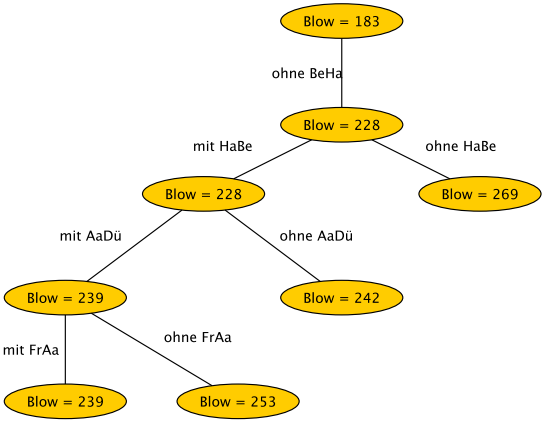


	Aa	Ba	Fr	Ha	Mü	Nü	St
Ba	24	∞	6	26	10	16	0
Be	14	36	11	∞	16	0	19
Dü	∞	33	9	0	50	31	28
Fr	0	6	∞	0	20	2	0
Mü	41	12	22	33	∞	0	5
Nü	24	18	4	16	0	∞	2
St	21	0	0	21	3	0	∞

Als nächstes betrachten wir die Kante FrAa.

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

Also mit FrAa, $\Rightarrow$ ohne DüFr.

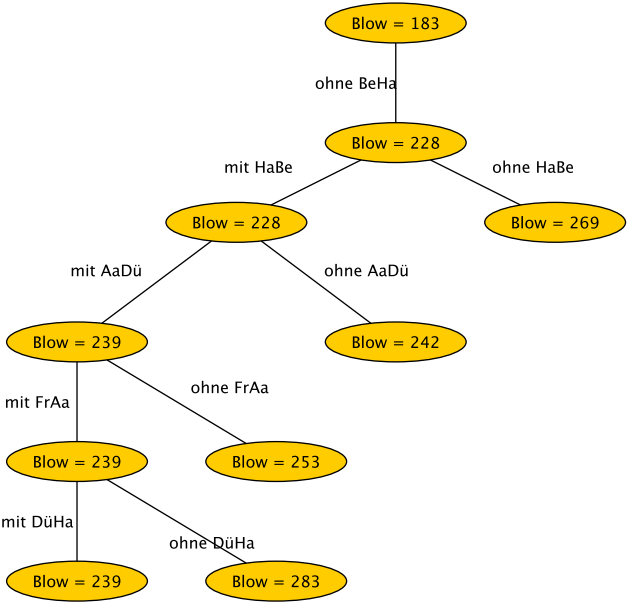


	Ba	Fr	Ha	Mü	Nü	St
Ba	∞	6	26	10	16	0
Be	36	11	∞	16	0	19
Dü	33	∞	0	50	31	28
Mü	12	22	33	∞	0	5
Nü	18	4	16	0	∞	2
St	0	0	21	3	0	∞

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.



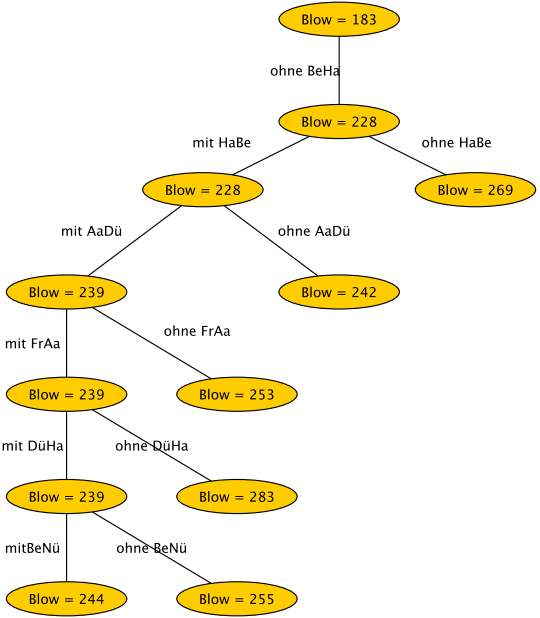
	Ba	Fr	Mü	Nü	St
Ba	∞	6	10	16	0
Be	36	∞	16	0	19
Mü	12	22	∞	0	5
Nü	18	4	0	∞	2
St	0	0	3	0	∞

Nun die Kante BeNü.

Ohne BeNü:
$B_{low} = 239 + 16 = 255 \geq 250$.
Ausgelotet!

Mit BeNü: $B_{low} = 239 + 5 = 244$.

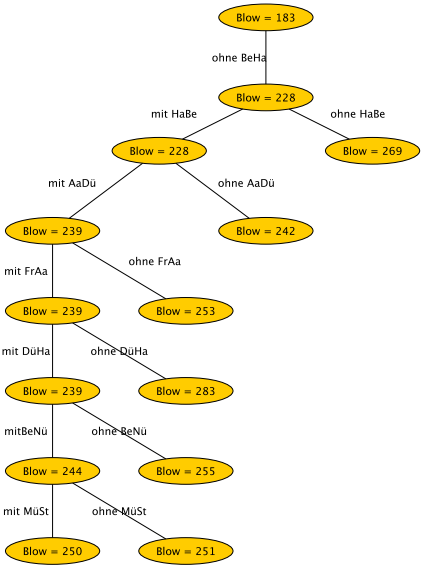
$\Rightarrow$ ohne NüFr.



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

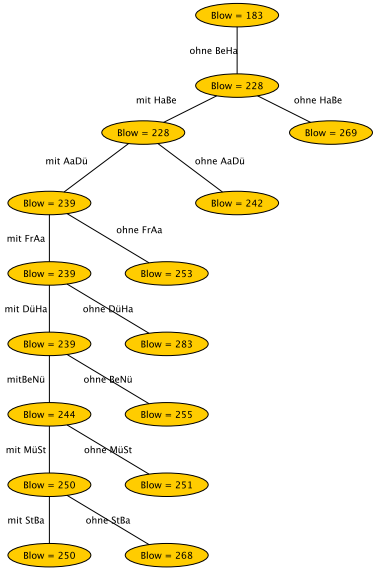


	Ba	Fr	Mü
Ba	∞	0	4
Nü	18	∞	0
St	0	0	3

Ohne StBa:

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

Ausgelotet!

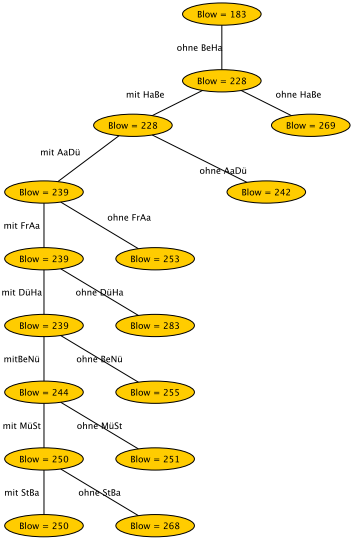


	Fr	Mü
Ba	0	4
Nü	∞	0

Wähle BaFr und NüMü.

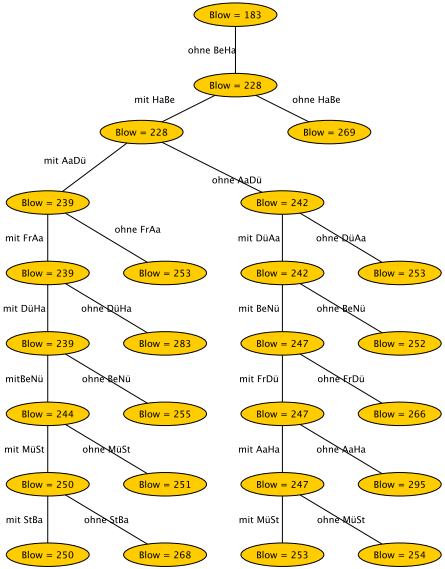
$\Rightarrow$ bekannte Tour mit $B_{up} = 250$.

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 ✎


Blow = 183
ohne BeHa
Blow = 228
mit HaBe
ohne HaBe
Blow = 228
Blow = 269
mit AaDü
ohne AaDü
Blow = 239
Blow = 242
mit FrAa
ohne FrAa
mit DüAa
ohne DüAa
Blow = 239
Blow = 253
Blow = 242
Blow = 253
mit DüHa
ohne DüHa
mit BeNü
ohne BeNü
Blow = 239
Blow = 283
Blow = 247
Blow = 252
mitBeNü
ohne BeNü
mit FrDü
ohne FrDü
Blow = 244
Blow = 255
Blow = 247
Blow = 266
mit MüSt
ohne MüSt
mit AaHa
ohne AaHa
Blow = 250
Blow = 251
Blow = 247
Blow = 295
mit StBa
ohne StBa
mit MüSt
ohne MüSt
Blow = 250
Blow = 268
Blow = 253
Blow = 254

Zusammenfassung

- Branch-and-Bound: Problem aufteilen in Teilprobleme und Teilprobleme lösen
- Bildung der Teilprobleme durch Fixierung von Variablen
- Verwendung von oberen und unteren Schranken, um Teilprobleme möglichst schnell ausloten zu können
- Maximierung: obere Schranken durch Relaxationen, untere durch Heuristiken