Lineare und Diskrete Optimierung

Manuskript zur Vorlesung

WS 1996/1997

Uwe Zimmermann
Abteilung Mathematische Optimierung
Technische Universität Braunschweig

Braunschweig, den 1. Oktober 1996

Inhaltsverzeichnis

Kapitel II

Diskrete Optimierung

1 Transportprobleme

Beim allgemeinen *Transportproblem* ist der Bedarf b_j eines homogenen Transportgutes in n Bedarfsorten mit dem Angebot a_i in m Angebotsorten kostenminimal zu decken. Wir gehen zunächst davon aus, daß jeder Bedarfsort von jedem Angebotsort aus zu erreichen ist und modellieren das Problem mit Hilfe eines vollständigen, bipartiten gerichteten *Transportgraphen.*

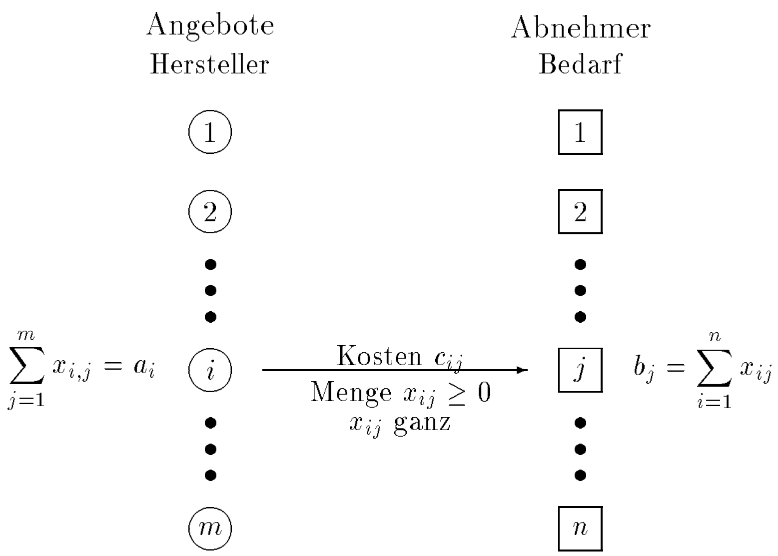


Abbildung II.1: Transportgraph

Zielfunktion: min $\sum_{i,j} c_{ij} x_{ij}$. Notwendig $\sum_i a_i = \sum_j b_j$, d.h. Angebot=Bedarf.

Knoten-Kanten-Inzidenzmatrix $A(G)$

$G = (V, E)$ gerichteter Graph. $A(G) := (a_{ie})_{i \in V, e \in E}$ mit

$$a_{ie} := \begin{cases} -1 & e = ij \\ 0 & \text{sonst} \\ 1 & e = ji \end{cases}$$

Beispiel Transportgraph:

$$\begin{array}{c|ccc|ccc|c|ccc|}
 & x_{11} & x_{12} \cdots & x_{1n} & x_{21} & x_{22} \cdots & x_{2n} & \cdots & x_{m1} & x_{m2} \cdots & x_{mn} \\
\hline
1 & -1 & -1 \cdots & -1 & & & & & & & \\
2 & & & & -1 & -1 \cdots & -1 & & & & \\
\vdots & & & & & & & \cdots & & & \\
m & & & & & & & & -1 & -1 \cdots & -1 \\
\hline
\boxed{1} & 1 & & & 1 & & & & 1 & & \\
\boxed{2} & & 1 & & & 1 & & & & 1 & \\
\vdots & & \ddots & & & \ddots & & \ddots & & \ddots & \\
\boxed{n} & & & 1 & & & 1 & & & & 1 \\
\hline
\end{array}$$

Abbildung II.2: Knoten-Kanten-Inzidenzmatrix für den Transportgraph

$Ax = \begin{pmatrix} -a \\ b \end{pmatrix}$, $x \geq 0$. A ist nicht von maximalem Rang. $1^T A = 0^T \rightsquigarrow \mathrm{rg} A \leq m+n-1$.

Lemma 1.1 *$G = (V, E)$ gerichteter, zusammenhängender Graph. $|V| = r$. Dann gilt $\mathrm{rg} A(G) = r - 1$.*

Beweis. Offensichtlich sind die Zeilen von $A(G)$ linear abhängig: $1^T A(G) = 0^T \rightsquigarrow \mathrm{rg} A(G) \leq r - 1$. Sei (V, B) Gerüst von G. $A_B = A(G)_B$ ist dann eine $r \times (r-1)$–Matrix. Beweisidee: Permutiere die Zeilen und Spalten von A_B so, daß die $(r-1) \times (r-1)$–Teilmatrix der ersten $r - 1$ Zeilen eine untere Dreiecksmatrix bildet, deren Diagonalelemente $+1$ oder -1 sind.

per Induktion erhalten wir

Somit gilt $\mathrm{rg} A = r - 1$. A_B zu Gerüst besitzt maximalen Rang.

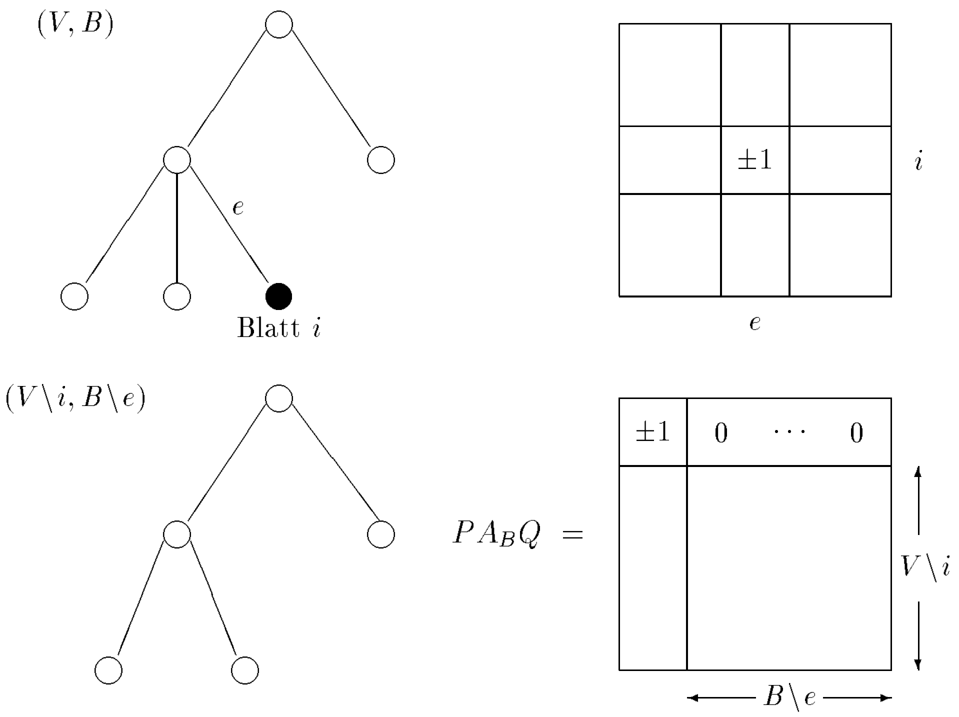


Abbildung II.3: zum Lemma

Gilt auch die Umkehrung? (V,S) sei kein Gerüst, $S \subset E, |S| = r-1 \rightsquigarrow (V,S)$ enthält einen Kreis (V,K).

A_k ist $k \times k$ Matrix.

$$x_e = \begin{cases} 1 & e \text{ in Richtung des Kreises} \\ -1 & \text{sonst} \end{cases}$$

$A_K x_K = 0$. Also sind die Spalten von A_S linear anhängig. Somit ist $\text{rg}A_S < r-1$. □

Satz 1.2 *$G = (V,E)$ gerichteter, zusammenhängender Graph, $|V| = r$. Sei $B \subset E$ mit $|B| = r-1$. Dann gilt: (V,B) ist ein Gerüst $\Leftrightarrow$ $\text{rg}A_B = r-1$.*

Basislösungen des TP

$$\min\left\{c^T x | Ax = \begin{pmatrix} -a \\ b \end{pmatrix}, x \geq 0\right\} \tag{1.1}$$

x ist eine BL genau dann, wenn für ein Gerüst (V,B) gilt

$$A_B x_B = \begin{pmatrix} -a \\ b \end{pmatrix}, x_N = 0$$

x ist zulässige BL, falls zusätzlich $x_B \geq 0$.

$$\tilde{P} A_B \tilde{Q} \;=\; \begin{array}{|ccccc|} \hline \boxed{\pm 1} & 0 & \cdots & & 0 \\ & \boxed{\pm 1} & 0 & \cdots & 0 \\ & & \boxed{\pm 1} & \cdots & 0 \\ & & & \ddots & 0 \\ & & & & \boxed{\pm 1} \\ \hline & & & & \\ \hline \end{array} \;\rightsquigarrow\; \operatorname{rg} A_B = r-1.$$

Abbildung II.4: Die Inzidenzmatrix nach $r-1$ Schritten

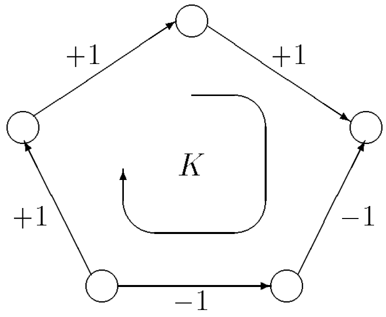


Abbildung II.5: Der Kreis (V, K)

Duales Problem

$$\max\{-a^T u + b^T v \mid -u_i + v_j \leq c_{ij} \quad \forall\, ij\}$$

Optimalitätstest des SV

1. Löse $(u^T v^T) A_B := c_B^T$
2. $\tilde{c}_N^T = c_N^T - (u^T v^T) A_N \geq 0 \rightsquigarrow B$ ist optimales Gerüst.

Lösung von 1

$$-u_i + v_j = c_{ij} \quad \forall\, ij \in B$$

$r-1$ Gleichungen für r Unbekannte. Ansatz $u_1 = 0$. Lösung längs Baumes ausgehend von beliebig gewählter Wurzel stets möglich.

Beispiel:

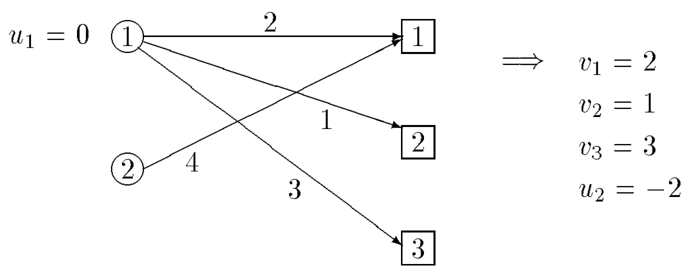


Abbildung II.6: Beispiel mit 2 Herstellern und 3 Abnehmern

Berechnung von 2

$$-u_i + v_j \le c_{ij} \quad \forall\, ij \in N$$

duale Zulässigkeit.

Basis (Gerüst-) Wechsel

Wähle eine Kante $rs \in N$ mit

$$\tilde{c}_{rs} = \min\{c_{ij} + u_i - v_j | ij \in N\} < 0$$

Der Graph $(V, B \cup \{rs\})$ enthält einen Kreis K.

Beispiel:

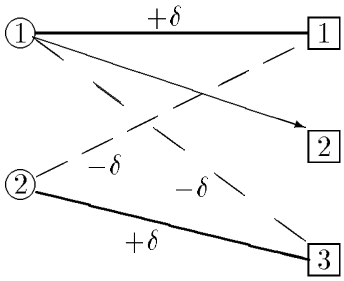


Abbildung II.7: (blaue Kanten fett, rote Kanten gestrichelt, rs blau)

$|K|$ gerade, da G bipartit, $\delta := \min\{x_{ij} | ij \text{ rot}\}$

$$0 \le x_{ij}(\delta) := \begin{cases} x_{ij} + \delta & ij \text{ blau} \\ x_{ij} - \delta & ij \text{ rot} \\ x_{ij} & \text{andernfalls} \end{cases}, \; Ax(\delta) = \begin{pmatrix} -a \\ b \end{pmatrix}$$

Eliminiere rote Kante pq mit $x_{pq} = \delta$, $(x_{pq}(\delta) = 0)$. Somit ist $B \cup \{rs\} \setminus \{pq\}$ ein Gerüst.

Bemerkung: Falls Startlösung ganzzahlig, so ist jede folgende BL ganzzahlig.

Bestimmung der Startlösung

Algorithmus 1.1

0. $B = \emptyset$
1. Wähle ij Setze $x_{ij} := \min\{a_i, b_j\}$; $B := B \cup \{ij\}$
2. Reduziere den Graphen
   (a) $a_i < b_j$: Streiche Knoten i, $b_j = b_j - a_i$
   (b) $a_i > b_j$: Streiche Knoten $\boxed{\text{j}}$, $a_i = a_i - b_j$
   (c) $a_i = b_j$:
      i. Streiche Knoten i, falls weitere Angebotsknoten existieren.
      ii. Andernfalls streiche Knoten $\boxed{\text{j}}$, falls weitere Nachfrageknoten existieren.
      iii. ij war letzte Kante. Stop.
3. Gehe nach 1.

Bemerkung: Algorithmus 1.1 liefert ein Gerüst mit zugehöriger BL.
Beweis.

1. Pro Iteration wird genau ein Knoten gestrichen (mit allen zugehöhrigen Kanten). $n + m - 1$ Kanten gewählt bei Abbruch.
2. B ist kreisfrei. Annahme: B enthält einen Kreis K. Sei $ij \in K$ die erste vom Algorithmus 1.1 in K ausgewählte Kante. Anschließend wurde i oder $\boxed{\text{j}}$ gestrichen, oBdA etwa i. D.h. keine weitere Kante in K kann mit i inzidieren. Widerspruch

Also wegen 1 und 2 ist B ein Gerüst. x BL wegen Restriktionen. □

Auswahlregeln

Sei $M \hat{=}$ ungestrichene Angebotsknoten, $N \hat{=}$ ungestrichene Nachfrageknoten.

1. **Nordwesteckenregel:** $i = \min M$, $j = \min N$.
2. **Regel der geringsten Kosten:** $c_{ij} := \min\{c_{kl} | k \in M, l \in N\}$.
3. **Modifizierte Zeilenminimumsregel:** Zeilen seinen zyklisch geordnet: $1, 2, \dots, m, 1, 2, \dots$. Start in Zeile 1, $i = 1$. Nach Wahl einer Kante in Zeile i wird in der Folgeiteration in der zyklisch nächsten Zeile μ die Kante $\mu\nu$ kostenminimal gewählt:

$$c_{\mu\nu} := \min\{c_{\mu j} | j \in N\}$$

Listenspeicherung von Bäumen

(V, B) Baum, $w \in V$ Wurzel. Es existiert ein Weg $W(j)$ von w nach $j \neq w$.

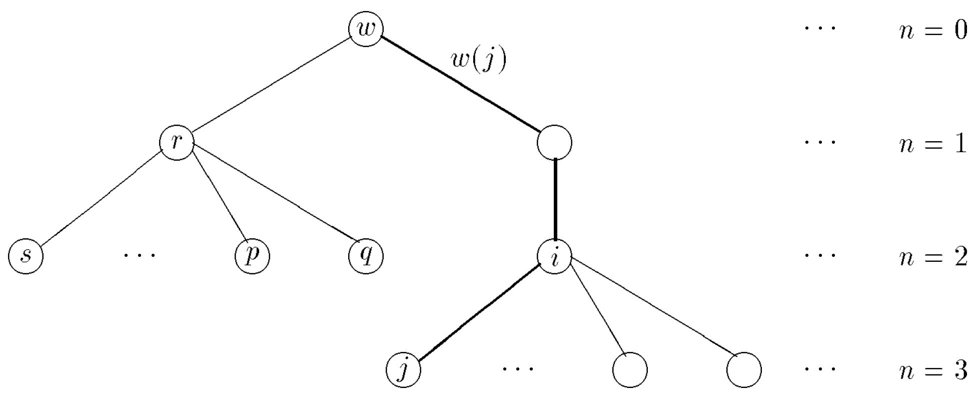


Abbildung II.8: Listenspeicherung von Bäumen

1. **unmittelbare Vorgänger** $p(w) := 0$, $p(j) := i$ (von j auf $W(j)$)
2. **unmittelbare Nachfolger** $s(r) := s$ (bei mehreren, jeweils der linksaußen), $s(j) := 0$ (falls keiner vorhanden).
3. **rechter Bruder** $b(p) := q$, $b(q) = 0$ (kein rechter Bruder).
4. **Niveau** $n(j) := 3(= |W(j)|)$, $n(w) := 0$.

Beispiel: Knoten durchnummerieren:

$$\{①, ②, ③, \boxed{1}, \boxed{2}, \boxed{3}, \boxed{4}\} \hat{=} \{1, 2, 3, 4, 5, 6, 7\}$$

		p	s	b	n	x	$\pi = (u, v)^T$
①	1	0	7	0	0	0	0
②	2	4	5	0	2	1	−3
③	3	5	0	0	4	4	−1
$\boxed{1}$	4	1	2	0	1	1	2
$\boxed{2}$	5	2	3	6	3	0	1
$\boxed{3}$	6	2	0	0	3	6	−1
$\boxed{4}$	7	1	0	4	1	3	1

Die Spalten 3–5 fixieren den Baum.

$$x(i) := \begin{cases} x_{ij} & \text{falls } j = p(i) > i \\ x_{ji} & \text{falls } j = p(i) < i \\ 0 & i = 1 \end{cases}$$

Optimalität

	π	2	1	-1	1
$\pi(m+j) \leq c_{1j} + \pi(1)$	0	$*$	3	4	$*$
$\pi(m+j) \leq c_{2j} + \pi(2)$	3	$*$	$*$	$*$	3

Bestimmung des von rs induzierten Kreises K

1. Bestimme den ersten gemeinsamen Vorgänger v von r und s. $n(v) \leq \min\{n(r), n(s)\}$

2. Dabei δ im Kreis bestimmen.

$$\delta = \min \left\{ x(\mu) | \mu = \begin{array}{c} r, p^2(r), \ldots \\ s, p^2(s), \ldots \end{array}, n(\mu) > n(v) \right\} =: x(t)$$

3. $p \in \{rs\}$ sei nach Streichen der Kante ut noch mit 1 verbunden. $q := \{r, s\} \setminus p$.

Änderung der Basislösung auf K

1. $x|_{K\setminus\{v,s\}} := x|_{K\setminus\{v,s\}} \pm \delta$.

2. Auf Teilkreis von q bis t $x(j)$ auf $x(p(j))$ verschieben.

Umhängen des Teilbaumes

Beispiel:

$$\begin{array}{|c|c|} \hline & b^T \\ \hline a & C \\ \hline \end{array} = \begin{array}{|c|cccc|} \hline & 2 & 4 & 6 & 3 \\ \hline 4 & 2 & 3 & 4 & 1 \\ 7 & 5 & 4 & 2 & 3 \\ 4 & 4 & 2 & 8 & 6 \\ \hline \end{array}$$

Bestimmung der Anfangslösung

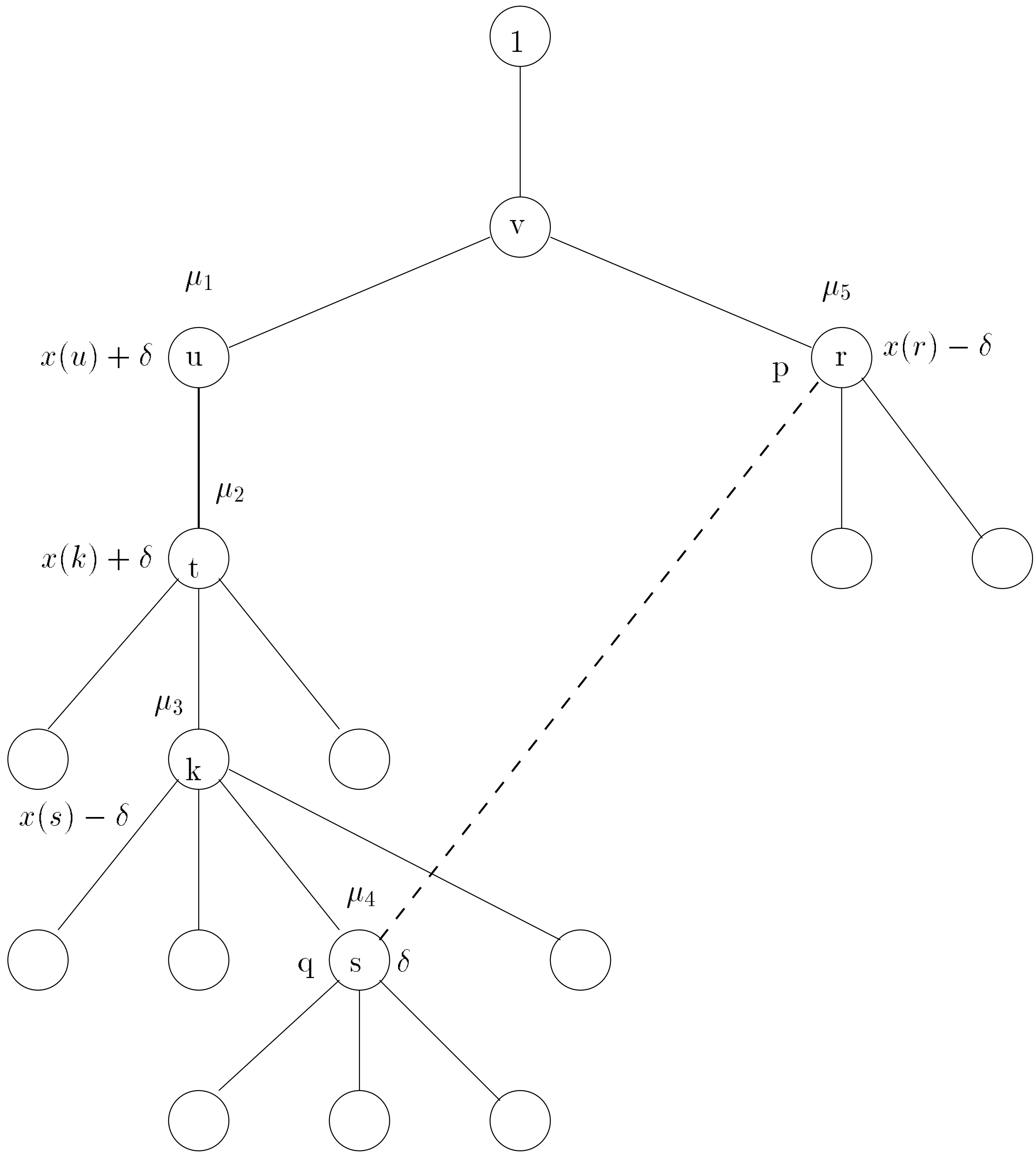


Abbildung II.9: Umhängen des Baumes

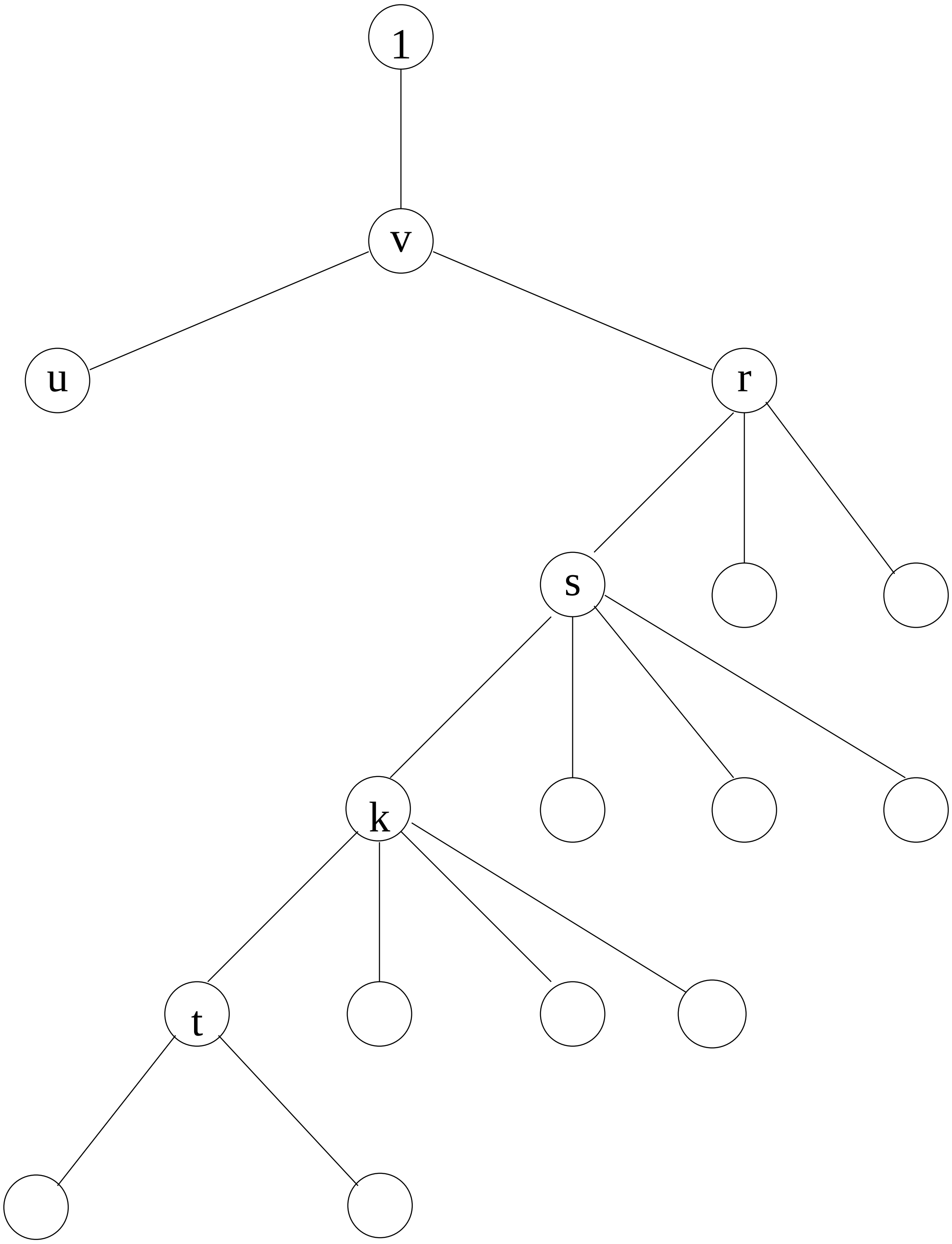


Abbildung II.10: Umgehängter Baum

1) Nordwesteckenregel

	1.	3.	5.	
2.	2	2		
4.		2	5	
6.			1	3

$c^T x = 4 + 6 + 8 + 10$
$+8 + 18 = 54$

2) Regel der geringsten Kosten

		5.	3.	1.
2.	1			3
6.	1	0	6	
4.		4		

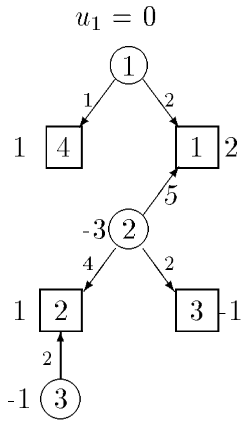
$c^T x = 2 + 3 + 5 + 0$
$+12 + 8 = 30$

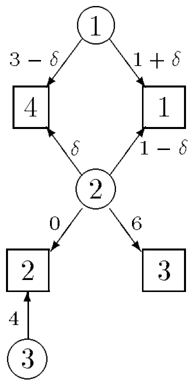
3) Modifizierte Zeilenminimumregel

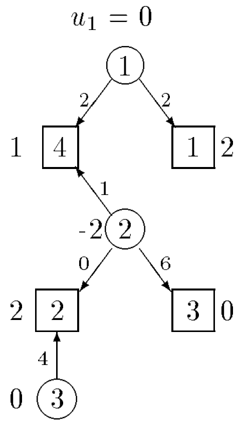
		5.	2.	1.
4.	1			3
6.	1	0	6	
3.		4		

$c^T x = 2 + 3 + 5 + 0$
$+12 + 8 = 30$

Optimierung der Lösung







Startgerüst B, $v_j = c_{ij} + u_i$
Optimalitätstest:
$v_j \leq c_{ij} + u_i \forall ij \in N$?
$v_4 = 1 > 3 - 3 = c_{24} + u_2$
$\Rightarrow rs = 24$

Basiswechsel:
$\delta = \min\{1, 3\} = 1$

Optimallösung,
Kantenbezifferung ist x_{ij}

Seien $u = \mu_1, \mu_2, \ldots, \mu_l = q, \mu_{l+1} = p$ die Knoten des Weges von u nach p über q im Kreis:

1. Nachfahren von u, t (zusammenschieben)

Für $\nu \in \{1, 2\}, \nu < l$:
Falls $s(\mu_\nu) = \mu_{\nu+1}$
setze $s(\mu_\nu) := b(\mu_{\nu+1})$
anderenfalls,
falls $b(\mu') = \mu_{\nu+1}$
setze $b(\mu') := b(\mu_{\nu+1})$.

2. Nachfahren der Zwischenknoten

Für $\nu \in \{3, \ldots, l-1\}$:
Falls $s(\mu_\nu) = \mu_{\nu+1}$
setze $b(\mu_{\nu-1}) := b(\mu_{\nu+1})$
anderenfalls,
falls $b(\mu') = \mu_{\nu+1}$
setze $b(\mu_{\nu-1}) := s(\mu_\nu)$
setze $b(\mu') := b(\mu_{\nu+1})$.

3. Nachfahren von r, s

Für $\nu \in \{l-1, l\}, \nu > 1$:
Setze $b(\mu_\nu) := s(\mu_{\nu+1})$

4. Unmittelbare Vorgänger / Nachfolger

Für $\nu \in \{2, \ldots, l\}$:
Setze $p(\mu_\nu) := \mu_{\nu+1}$,
setze $s(\mu_{\nu+1}) := \mu_\nu$.

Änderung der Niveaugrößen und der Dualvariablen

1. $n(\ldots)$ trivial rekursiv ab $n(s) := n(r) + 1$.

2. $\beta := \tilde{c}_{rs} = c_{rs} + \pi(r) - \pi(m+s) < 0$. Für alle Nachfolger von q (einschließlich):

$$\pi(i) := \begin{cases} \pi(i) - \beta & q = r \\ \pi(i) + \beta & q = s \end{cases}$$

Beispiel:

$$\begin{aligned} \beta &= c_{24} + u_2 - v_4 \\ &= 3 + (-3) - 1 = -1 \end{aligned}$$

$v_4 = 1 \; \boxed{4} \leftarrow ② \; {\scriptstyle u_2 = -3}. \; \underbrace{v_4}_{v_4'} = c_{24} + \underbrace{u_2 - \beta}_{u_2'}.$

Bei anderer Richtung als im Beispiel: $-3 \; ② \rightarrow \boxed{4} \; {\scriptstyle 1}. \; \underbrace{v_4 + \beta}_{v_4'} = c_{24} + \underbrace{u_2}_{u_2'}.$

Optimalitätskontrolle

$z_{opt} = -a^T u + b^T v = (-a \quad b)\pi = (-4, -7, -4, 2, 4, 6, 3)(0, -2, 0, 2, 2, 0, 1)^T = 29.$

Bemerkung: Variationen der Speichertechnik sind vielfältig, siehe z.B.

1. Kennigton/Helgason: Algorithms for Network Programming. Wiley, NY 1980.

2. Jensen/Barnes: Network Flow Programming. Wiley, NY 1980.

3. Domschke: Logistik, Transport. Oldenburg, München 1981.

Modifikationen

1. Probleme mit Überschuß: Lager: $\sum a_i > \sum b_j$, Lagerknoten mit Bedarf $b_{n+1} := \sum a_i - \sum b_j$ einfügen, Lagerkosten und Transport ins Lager in Kosten c berücksichtigen.

2. Probleme mit Defizit: analog.

3. Probleme mit fehlenden Transportverbindungen: Nur Startlösung kritisch, falls Kante ij nicht vorhanden. Hilfsphase einfügen: $c'_{ij} := \begin{cases} 0 & ij \text{ vorhanden} \\ 1 & ij \text{ nicht vorhanden} \end{cases}$

   Lösung dieser Phase sei x mit $c'^T x \begin{cases} = 0 & \text{ex. zul. Transportlösung} \\ > 0 & \text{ex. keine zul. Transportlösung} \end{cases}$

Engpaßprobleme

$$\bar{t} := \min\{\max_{x_{ij}>0} t_{ij} | x \in P\}$$

Anwendung:

1. Transportdauer $\hat{=} t_{ij}$.

2. Engpässe maximieren: $\max\{\min_{x_{ij}>0} \tilde{t}_{ij} | x \in P\}$ „assembly line balance“. $\tilde{t}_{ij}$ = Durchsatz am Fließband bei Einsatz von Personal i auf Platz j. $a_i = b_j = 1 \quad \forall\, i, j$. „Zuordnungsproblem“

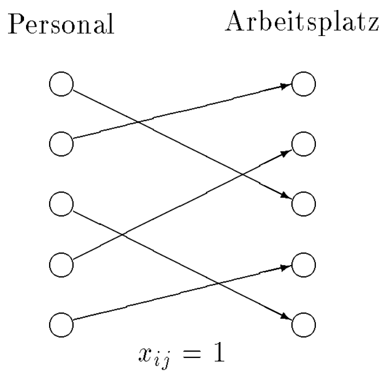


Abbildung II.11: Zuordnungsproblem

3. Verfeinerung von 1.

   $t(x) := \max_{x_{ij}>0} t_{ij} \rightsquigarrow$ minimale Transportzeit.

   $m(x) := \sum_{t_{ij}=t(x)} x_{ij} \rightsquigarrow$ minimiere Transportmenge mit maximaler Transportzeit.

Lexikographisches Engpaßproblem

$$\text{lexmin}\left\{\begin{pmatrix} t(x) \\ m(x) \end{pmatrix} \,\Big|\, x \in P\right\}$$

Hilfsproblem $P(t)$ zu $t \in \{t_{ij} | ij \in E\}$

$$z(t) := \min\{\sum_{i,j} c_{ij}(t) x_{ij} | x \in P\}$$

mit

$$c_{ij}(t) := \begin{cases} 0 & t > t_{ij} \\ 1 & t = t_{ij} \\ M & t < t_{ij} \end{cases}$$

mit hinreichend großem M. Z.B. $M \geq \sum_i a_i + 1$.

Lemma 1.3 *Sei x OL zu $P(t)$. Dann gilt*

1. $z(t) = 0 \Leftrightarrow \bar{t} < t$.
2. $0 < z(t) < M \Leftrightarrow \bar{t} = t$.
3. $z(t) \geq M \Leftrightarrow \bar{t} > t$.

Beweis. trivial, für hinreichend großes M.

Korollar 1.4 *(Erweiterte) Engpaßprobleme besitzen stets eine optimale Basislösung.*

Das Optimum wird in einer Ecke angenommen.

Verfahren für (Erweiterte) Engpaßprobleme

1. Bestimme BL $x \in P$. $t := t(x)$.
2. Bestimme ausgehend von x die optimale BL $\tilde{x}$ von $P(t)$ mit Wert $z(t)$.
3. Falls $z(t) > 0$ stop. ($\tilde{x}$ löst das Erweiterte Engpassproblem.)
   Andernfalls $t := t(\tilde{x})$; $x := \tilde{x}$ und gehe nach 2.

Untere Schranke $\underline{t}$ zum Engpaßproblem

1. OL für $m = 1$: OBdA $t_{11} \leq t_{12} \leq \ldots \leq t_{1n}$ a_1 sukzessive über $11, 12, \ldots, 1n$ versenden.
   $$\bar{t}_1 := \min\{t_{1\mu} | \sum_{\nu=1}^{\mu} b_\nu \geq a_1\}$$
   Analog $\bar{t}_i$ für $i = 2, \ldots, m$, $\bar{\bar{t}}_j$ für $j = 1, \ldots, n$.

2.

$$\underline{t} := \max\{\bar{t}_1, \ldots, \bar{t}_m, \bar{\bar{t}}_1, \ldots, \bar{\bar{t}}_n\} \leq \bar{t}$$

Bemerkung:

1. Geschickte, effiziente Bestimmung von $\underline{t}$ ist bekannt.
2. Bei gleichverteilten t_{ij} ist $\underline{t}$ in der Praxis in 94% der Beispiele optimal.

Startlösung

1. $z_i := |\{ij | t_{ij} \leq \underline{t}\}|$, $s_j := |\{ij | t_{ij} \leq \underline{t}\}| \quad \forall\, ij$
2. Auswahlregel im Algorithmus 1.1:

   $ZS := \{z_\mu, s_\nu | z_\mu > 0, s_\nu > 0, \mu \in M, \nu \in N\}$, $\alpha := \min ZS$.

   Falls $\alpha = z_i$ für ein $i \in M$ wähle ij mit $t_{ij} \leq \underline{t}$.

   Andernfalls, für $\alpha = s_j$ für ein $j \in N$ wähle ij mit $t_{ij} \leq \underline{t}$.
3. Sobald $ZS = \emptyset$, restliche Auswahl mit modifizierter Zeilenminimumsregel treffen. Anderenfalls gehe nach 1.

Bemerkung:

1. Teillösung $\tilde{x}$ nach 1/2 erfüllt $t(\tilde{x}) \leq \underline{t}$.
2. $\tilde{x} \in P \rightsquigarrow \tilde{x}$ ist optimal für das Engpaßproblem, nicht unbedingt für das erweiterte Engpaßproblem.
3. Das Verfahren benötigt erfahrungsgemäß etwa das zweifache der Rechenzeit des besten bekannten Verfahrens.

Beispiel:

	b	
a	T	$\bar{t}$
	$\bar{\bar{t}}$	

	2	4	6	3	
4	2	3	4	1	2
7	5	4	2	3	3
4	4	2	8	6	2
	2	2	2	1	

$\rightsquigarrow \underline{t} = 3$

2	0	$*$	2	3	3	2	1	$*$
$*$	$*$	6	1	2	2	2	2	2
$*$	4	$*$	$*$	1	$*$	$*$	$*$	$*$

1	2	1	2		
1	1	1	2	$\alpha_1 = 1 = z_3$	$\alpha_4 = 1 = z_1$
$*$	1	1	2	$\alpha_2 = 1 = s_1$	$\alpha_5 = 1 = s_3$
$*$	$*$	1	2	$\alpha_3 = 1 = s_2$	$\alpha_6 = 1 = z_2$
$*$	$*$	1	1		

$\leadsto \tilde{x} = x$ zulässig. Also Engpaßproblem gelöst. $t(\tilde{x}) = 3, m(\tilde{x}) = 1$. D.h. in diesem Beispiel ist auch das erweiterte Engpaßproblem gelöst.

2 Flüsse

$G(V,E)$ gerichteter Graph mit $l \leq c : E \to \boldsymbol{R}$, l untere, c obere Kapazitäten der Kanten.

Bezeichnungen

1. $l(E') := \sum_{e' \in E'} l(e')$, $E' \subseteq E$, $l(\emptyset) := 0$.
2. $\delta(V') := \{ij | i \in V', j \in V \setminus V'\}$, $V' \subseteq V$.
3. $\bar{V}' := V \setminus V'$, einelementige Mengen $\bar{v} = V \setminus \{v\}$.

Flüsse in G

$x : E \to \boldsymbol{R}$ mit

(2.1) Flußerhaltung $\quad x(\delta(\bar{v})) = x(\delta(v)) \quad \forall\, v \in V$

(2.2) Zulässigkeit $\quad l \leq x \leq c$

$P :=$ Menge der zulässigen Flüsse ist ein Polyeder.

Lemma 2.1 *$a : E \to \boldsymbol{R}$, Der Rand von a sei definiert durch $\partial a(S) := a(\delta(\bar{S})) - a(\delta(S))$ für $S \subseteq V$. Dann gilt $\partial a(S) = \sum_{v \in S} \partial a(v)$ (Modularität).*

Beweis. Induktiv über $|S|$. $S = \emptyset \ \surd$. $v \notin S, S \leadsto S \cup \{v\}$

$$\begin{array}{ccccc} \partial a(S \cup v) & = & \partial a(S) & + & \partial a(v) \\ (3) & & (1) & & -(1) \\ (2) & & (2) & & (3) \end{array}$$

□

Korollar 2.2 *$x : E \to \boldsymbol{R}$. Dann gilt (2.1) genau dann, wenn $\partial x(S) = 0 \quad \forall\, S \subset V$.*

Beweis. trivial mit Lemma 2.1. □

Lemma 2.3 *Ist $P \neq \emptyset$, so gilt $l(\delta(\bar{S})) \leq c(\delta(S)) \quad \forall\, S \subseteq V$.*

Beweis. Sei $x \in P$. $l(\delta(\bar{S})) \leq x(\delta(\bar{S})) = x(\delta(S)) \leq c(\delta(S))$. □

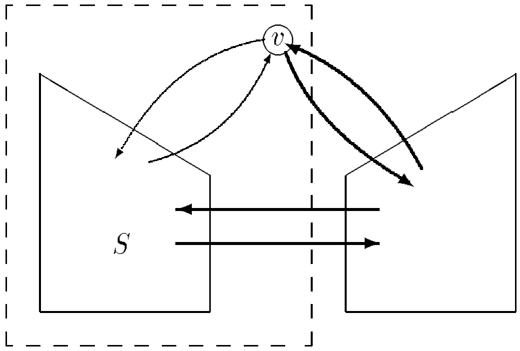


Abbildung II.12: zum Lemma

Inkrementgraph zu $x \in P$

$G_x := (V, E_x)$ mit

$$E_x := \{ij| \quad \underbrace{\overbrace{l(ji) < x(ji)}^{E_-}}_{0<c_x(ij):=x(ji)-l(ji)} \quad , \quad \underbrace{\overbrace{x(ij) < c(ij)}^{E_+}}_{0<c_x(ij):=c(ij)-x(ij)} \quad \},\ l_x = 0$$

Bemerkung: Zur Vermeidung von parallelen Kanten wird angenommen, daß G antisymmetrisch ist, d.h. $\neg(ij \in E \wedge ji \in E)$.

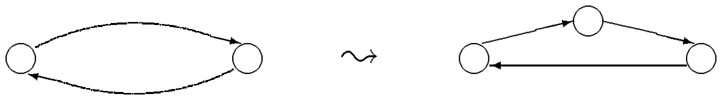


Abbildung II.13: G antisymmetrisch

Bezeichnungen:

1. Wege, Kreise, usw. im Inkrementgraphen heißen „Inkrementwege“, usw.
2. Kapazität eines Inkrementweges W: $c_x(W) := \min\{c_x(ij)|ij \in W\}$.
3. Inkrementfluß $\Delta x[C]$ auf einem Inkrementkreis C:

$$\Delta x(ij) := \begin{cases} c_x(C) & ij \in C \\ 0 & \text{andernfalls} \end{cases}$$

   Konform: $\Delta x(ij)\Delta x(ji) = 0$, wobei $\Delta x(e) := 0$ für $e \notin E_x$.

Flußänderungen

Ist Δx ein Inkrementfluß in G_x, so sei

$$(x \oplus \Delta x)(ij) := x(ij) + \Delta x(ij) - \Delta x(ji)$$

wobei $\Delta x(e) \hat{=} 0$ für $e \notin E_x$.

Lemma 2.4 *1. $x \oplus \Delta x \in P$*

2. Für alle $x, y \in P$ existiert in G_x ein konformes Δx mit $y = x \oplus \Delta x$.

Beweis.

1. zulässig $\surd$, da $x, \Delta x$ Flüsse gilt die Flußerhaltung.
2. $$\Delta x(ij) := \begin{cases} y(ij) - x(ij) & y(ij) > x(ij) \\ x(ji) - y(ji) & y(ji) < x(ji) \\ 0 & \text{andernfalls} \end{cases}$$

   zulässig in G_x, Fluß, da x, y Flüsse. □

Maximierung von Flußwerten

Sei $e \in E$.

(2.3) $$z_* := \max\{x(e) | \underbrace{x \text{ zulässiger Fluß in } G}_{P}\}$$

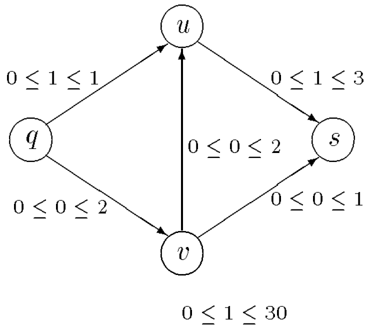
Algorithmus 2.1 (Kürzeste Wege Methode)
$e = sq$. $x \in P$

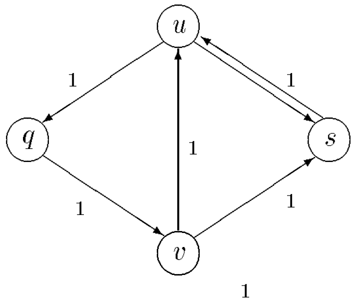
1. Falls $x(e) = c(e)$, stop.
2. Bestimme einen $\text{KW}_{q \rightsquigarrow s} W$ in $G_x \setminus qs$. Stop, falls nicht vorhanden.
3. $x := x \oplus \Delta x[W \cup e]$, gehe nach 1.

Bemerkung:

1. KW-Methode generiert eine Flußfolge mit streng isotonen $x(e)$. Falls l, c und Start x ganz, so ist x ganz. Abbruch nach maximal $c(e) - l(e)$ Iterationen.
2. Edmonds/Karp 1971: $O(nm^2)$ Verfahren.

Beispiel:





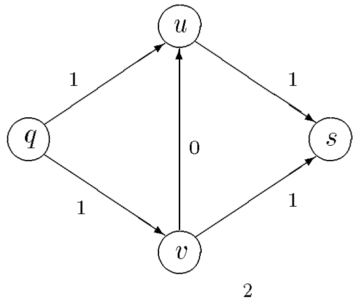
zulässiger Fluß | Inkrementfluß Δx

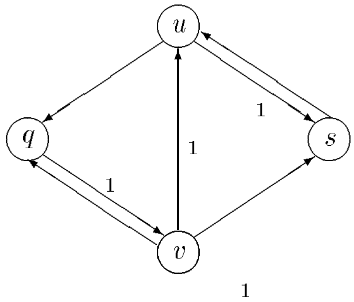
$l \le x \le c$ | $0 \le \Delta x \le c_x$

$\Rightarrow x \oplus \Delta x = 0$

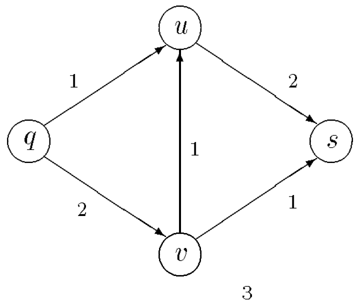
1. Iteration: $e = sq$

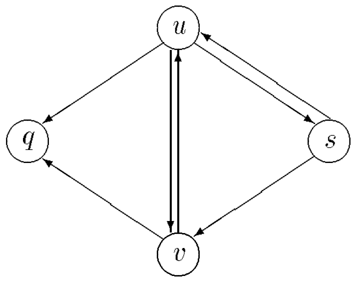
$x \oplus \Delta x$





$x \oplus \Delta x$





kein Weg von q nach s!
STOP

$q - s$-Schnitte

$q \in S,\ s \notin S,\ S \subseteq V$. Dann heißt S ein $q-s$-Schnitt mit Wert

$$z(S) := c(\delta(S)) - l(\delta(\bar{S}) \setminus sq).$$

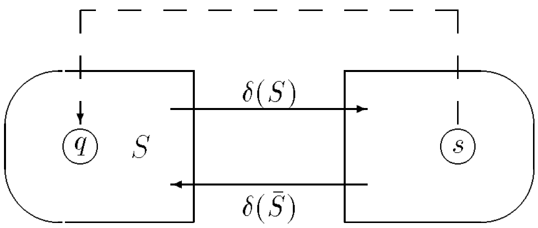


Abbildung II.14: $q-s$–Schnitt S

Folgerung:

1. $x(sq) \leq z(S) \quad \forall\, q-s$–Schnitte S.
2. $z_* \leq t_* := \min_{S\ q-s\text{–Schnitt}} z(S)$.

Beweis. zu 1.

$$\begin{aligned} x(sq) - z(S) &= x(sq) + l(\delta(\bar{S}) \setminus sq) - c(\delta(S)) \\ &\leq x(\delta(\bar{S})) - x(\delta(S)) = \partial x(S) = 0 \end{aligned}$$

□

Satz 2.5 *Sei $P \neq \emptyset$. Dann gilt $z_* = \min(t_*, c(e))$.*

Beweis. Nach Folgerung gilt "$\leq$". Sei x optimale Lösung zu (2.3): $z_* = x(e)$.

1.Fall $x(e) = c(e) \rightsquigarrow$ "$\geq$".

2.Fall $x(e) < c(e)$, $G_x \setminus qs$ enthält keinen Weg von q nach s. $S := \{v | v$ von q aus in $G_x \setminus qs$ erreichbar$\}$.

$$x(e) = x(\delta(S)) - x(\delta(\bar{S}) \setminus sq) = c(\delta(S)) - l(\delta(\bar{S}) \setminus sq) = z(S) \geq t_*.$$ □

Korollar 2.6 (Satz von Ford und Fulkerson) *$l \equiv 0$, $c(e) \geq c(E \setminus e)$. Dann gilt*

$$\max_{x \text{ zul. Fluß}} x(e) = \min_{S\ q-s\text{–Schnitt}} c(\delta(S))$$

Beweis. $l \equiv 0 \rightsquigarrow x \equiv 0 \in P$. $c(e) \geq c(E \setminus e) \rightsquigarrow c(e) \geq t_*$. $z(S) = c(\delta(S))$. □

Bemerkung: Die kürzeste-Wege-Methode liefert bei Abbruch einen "maximalen" Fluß x und einen "minimalen" Schnitt.

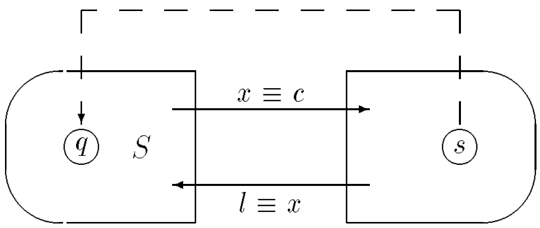


Abbildung II.15: zum Beweis

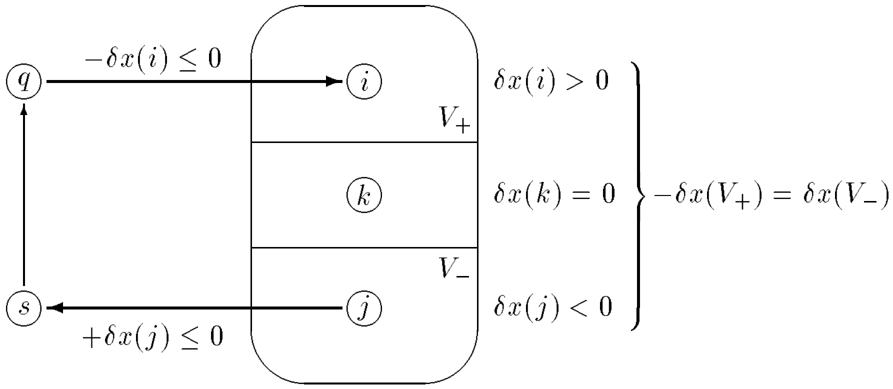


Abbildung II.16: Der Hilfgraph $\tilde{G}$

Satz 2.7 (Hoffmans Existenzbedingung) *$P \neq \emptyset$ genau dann, wenn $l(\delta(\bar{S})) \leq c(\delta(S)) \quad \forall S \subset V$.*

Beweis. Notwendig nach Lemma 2.3. Hinreichend: Idee: Rückführung auf Flußmaximierung in einem Hilfsgraphen. Start mit x, z.B. $x \equiv l$.

obere Kapazitäten auf neuen Kanten $\equiv 0$.

untere Kapazitäten auf neuen Kanten $\equiv -\infty$.

$z_* \equiv 0 \Leftrightarrow P \neq \emptyset$. Nach Satz 2.5 zu zeigen $t_* \geq 0$ in $\tilde{G}$. Sei S ein $q-s$–Schnitt in $\tilde{G}$.

blaue Kanten nur in $\tilde{\delta}(S)$. $c(\tilde{\delta}(S)) = c(\delta(T))$, $l(\tilde{\delta}(\bar{S}) \setminus sq) = l(\delta(V \setminus T) \rightsquigarrow \tilde{z}(S) = c(\delta(T)) - l(\delta(V \setminus T)) \geq 0$. □

Bemerkung: Startfluß in KW-Methode durch Anwendung von Algorithmus 2.1 auf $\tilde{G}$. Ein "maximaler" Fluß ergibt sich in $O(nm^2)$.

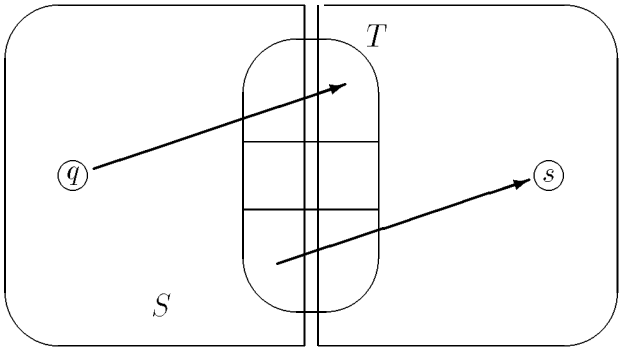


Abbildung II.17: Ein $q-s$–Schnitt in $\tilde{G}$

Der Schichtgraph S_x in G_x

$e = sq$. $l \hat{=}$ Entfernung der Senke s von q in $G_x \setminus qs$.

$$
\begin{array}{rcl}
S_x & := & (V_0 \,\dot\cup\, V_1 \,\dot\cup\, \ldots \,\dot\cup\, V_l, E_0 \,\dot\cup\, E_1 \,\dot\cup\, \ldots \,\dot\cup\, E_l) \text{ mit} \\
V_j & := & \text{Knoten mit Entfernung } j \text{ von } q \text{ in } G_x \setminus qs.\ 0 \leq j < l \\
V_l & := & \{s\} \\
E_j & := & \text{Kanten von } V_j \text{ nach } V_{j+1} \text{ in } G_x.\ 0 \leq j < l \\
E_l & := & \{sq\}
\end{array}
$$

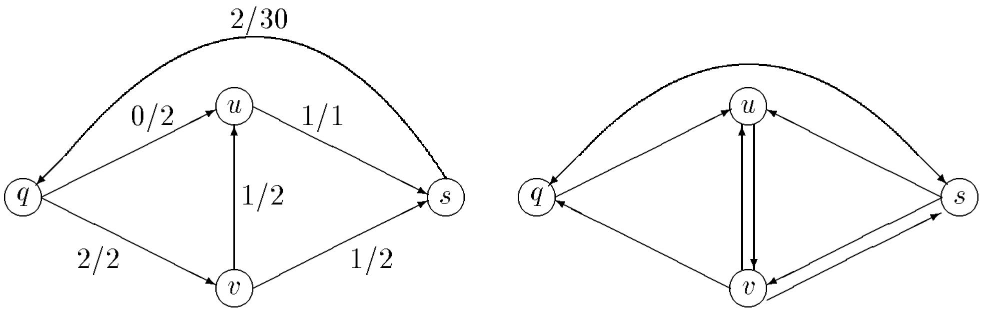


Abbildung II.18: Ein Beispiel für einen Schichtgraphen

Bemerkung: Jeder Weg von q nach s in G_x der Länge l liegt in S_x. Jeder Weg von q nach s in S_x hat Länge l.

Maximaler Fluß bezüglich $e = sq$

Ein Fluß heißt maximal, falls jeder Weg von q nach s eine Kante enthält, für die Flußwert und Kapazität übereinstimmen.

Verfahren maximaler Flüsse

1. Sei $x \in P_C$, $k := 1$.
2. Enthält G_x keinen Weg von q nach s, dann stop (optimal).
   Sonst bestimme in S_x einen maximalen Inkrementfluß Δx.
3. $X := x \oplus \Delta x$, $k := k + 1$. Gehe nach 2.

Zur Entwicklung der Graphen

	Iteration k	Iteration $k+1$
Inkrementgraphen	G'	G''
Schichtgraphen	$S = (V', E')$	$S'' = (V'', E'')$
Knoten	$V' = V'_0 \dot\cup \ldots \dot\cup V'_{l'}$	$V'' = V''_0 \dot\cup \ldots \dot\cup V''_{l''}$
Fluß	$x, \Delta x$	

Lemma 2.8 *Sei $ij \in E''$, aber $ij \notin E'$. Falls $i \in V'_\lambda$, so gilt entweder ($j \in V'_\mu$ mit $\mu \leq \lambda$) oder ($j \notin V', \lambda = l' - 1$).*

Beweis. Sei $i \in V'_\lambda$, aber $j \notin V'_\mu$ für alle $\mu \leq \lambda$. Annahme: $j \in V'_\mu$ mit $\mu \geq \lambda + 1$, d.h. $ji \notin E'$. Falls $ij \in G' \rightsquigarrow \mu = \lambda + 1$ und $ij \in E'$. Widerspruch zu $ij \notin E'$, also $ij \notin G'$. Also x_{ij} unverändert in Iteration $k \rightsquigarrow ij \notin G''$. Widerspruch zu $ij \in E''$. Also $j \notin V'$. Wiederum $ij, ji \notin E'$, also x_{ij} wieder unverändert. Wäre $ij \notin G'$, so folgt abermals $ij \notin G''$. Also $ij \in G', j \notin V'$, daher $i \in V'_{l'-1}$ (vgl. Abbildung II.19) □

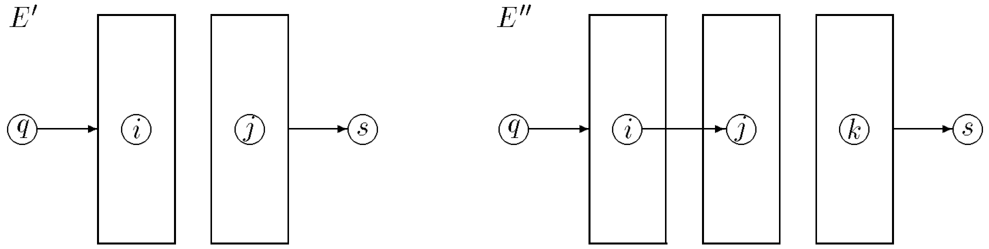


Abbildung II.19: j hat Entfernung l' von q

Lemma 2.9 *Die Entfernung des Knotens s von q in S_x wird in jeder (nicht abbrechenden) Iteration echt vergrößert.*

Beweis. Zu zeigen: $l'' > l'$. Sei $W_{q \leadsto s}$ Weg in S''.

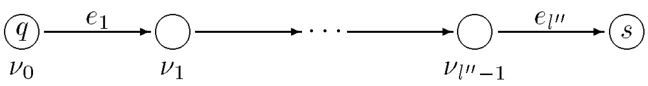


Abbildung II.20: Weg W in S''

Es gilt: Falls $\nu_0, \dots, \nu_a \in V'$, so gilt $\nu_a \in V'_b$ mit $b \leq a$. Induktion über a. $a = 0$ klar, da $\nu_0 = q \in V'$. $a \to a+1$: $\nu_a \in V'_b$ mit $b \leq a$. $\nu_{a+1} \in V'$. Falls $e_{a+1} \in E' \leadsto \mu = b+1 \leq a+1$. Andernfalls Lemma 2.8 anwenden $\nu_a \in V'_b, e_{a+1} \in E'', e_{a+1} \notin E', \nu_{a+1} \in V'_\mu \leadsto \mu \leq b \leq a < a+1$.

(2.4) Für jede in E' fehlende Kante liegt ν_a mindestens eine Schicht näher an q in S'

1.Fall $a = l''$ (alle Kanten von W in V'). Annahme: W Weg in $S' \leadsto \quad \exists\, e_j \in W$ mit $\Delta x(e_j) = c_x(e_j) \leadsto e_j \notin E''$. Widerspruch. Also fehlt mindestens eine Kante e_j in $E'' \overset{(2.4)}{\leadsto} s = \nu_a \in V'_b$ mit $l' = b < a = l''$.

2.Fall $a < l''$. Sei a minimal mit $\nu_a \in V'_\lambda$ aber $\nu_{a+1} \notin V'$, also $e_{a+1} \notin E'$ also Lemma 2.8 anwendbar. $l' - 1 = \lambda \leq a \leadsto l' \leq a+1, \nu_{a+1} \notin V' \leadsto \nu_{a+1} \neq s \leadsto a+1 < l'' \leadsto l' < l''$. $\square$

Korollar 2.10 *Das Verfahren maximaler Flüsse bricht nach höchstens $n-1$ Iterationen ab.*

Konstruktion maximaler Flüsse

Malhartrei, Pramodh Kumar, Maheswari (1978)

Beispiel:

Speichertechnik: Queue (Schlange) für zu bearbeitende Knoten.

Bemerkung:

1. Kante e in Schritt 6 wird nur dann eliminiert, wenn $\Delta y(e)$ durch keinen Fluß auf einem Weg von q nach s vergrößert werden kann.

2. Bei Abbruch y maximal.

3. Aufwand: majorisiert durch die Anzahl der Betrachtungen von Kanten.

   I Ausschöpfung der Kantenkapazität: Elimination $O(|E'|)$

   II teilweise Ausschöpfung der Kantenkapazität: höchstens einmal pro Knoten in Iteration von 1-6. Höchstens $|V'|$ Iterationen, wegen Elimination des Knotens minimaler Kapazität.

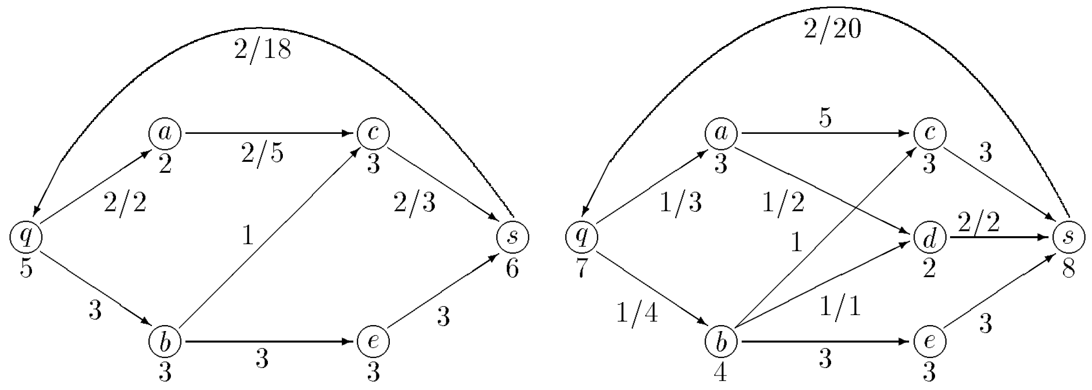


Abbildung II.21: Bestimmung maximaler Flüsse

Aufwand für Bestimmung eines maximalen Flusses: $O(|V'|^2)$.

Satz 2.11 *Das Verfahren maximaler Flüsse mit Hilfe von MPM bestimmt eine OL in $O(|V|^3)$ Operationen.*

Zusammenfassung:

Skizze einer Phase

Sei $G' = (V', E')$ ein Schichtgraph mit oberen Kapazitäten $c > 0$ und unteren Kapazitäten $l = 0$. Als Knotenkapazität definiert man $k(v) := \min\{c(\delta(v)), c(\delta(\bar{v}))\}$ für alle $v \in V'$.

Gesucht ist ein Fluß y in G', der $x(sq)$ maximiert.

1. Setze $y := 0$ (zulässiger Startfluß)
2. Bestimme $v \in V'$ mit minimaler Knotenkapazität. (Stop, falls $V' = \emptyset$)
3. Ausgehend von v, versende von Schicht zu Schicht $k(v)$ Flußeinheiten über s nach q.
4. Rückwärts ausgehend von v, hole von Schicht zu Schicht $k(v)$ Flußeinheiten von q.
5. Reduziere alle Kantenkapazitäten entsprechend dem in 3. und 4. bestimmten Fluß Δy.
6. Eliminiere alle Kanten mit Kapazität 0 und dann rekursiv alle Knoten mit Kapazität 0. (Streichen eines Knotens eliminiert alle adjazenten Kanten.)

7. Setze $y := y \oplus \Delta y$ und gehe nach 2.

Zusatzregel in 3. und 4.: Versenden und Holen von Flußeinheiten unter Ausnutzung vorhandener Kapazitäten durchführen; d.h. höchstens eine der genutzten Kanten zwischen zwei Schichten wird nur teilweise ausgeschöpft.

Skizze des MPM-Verfahrens

Sei $G = (V, E)$ ein Graph mit unteren und oberen Kapazitäten l, c mit $l \leq c$. $n := |V|$. Gesucht ist ein Fluß x in G, der $x(sq)$ maximiert.

1. Bestimme einen Startfluß x.

2. Bestimme den Inkrementgraphen $G_x = (V, E_x)$.
   Bestimme $d(v) := \min\{n, |W| \mid W$ ist kürzester Weg von q nach $v\}$ für alle $v \in V$.
   (n repräsentiert "Knoten nicht erreichbar", da der längste mögliche Weg $n-1$ Kanten enthalten kann)
   Falls $d(s) = n$, stop. (s ist unerreichbar, $x(sq)$ ist maximal.)

3. (Aufbau des Schichtgraphen)
   $l := d(s)$ (Entfernung von s)
   $V' := \{v \in V \mid d(v) < l\} \cup \{s\}$
   $E' := \{ij \in E_x \mid i, j \in V',\ \ d(j) = d(i) + 1\} \cup \{sq\}$.

4. Führe die oben beschriebene Phase zur Bestimmung von y durch.

5. Setze $x := x \oplus y$ und gehe nach 2.

0-1 Flußprobleme

$c \equiv 1, l \equiv 0 \rightsquigarrow c_x \equiv 1 \rightsquigarrow$ maximale Flüsse nur mittels Alternative **I** aufgebaut. Also werden maximale Flüsse in $O(|E'|)$ berechnet.

Lemma 2.12 $l \equiv 0, c \equiv 1$. *Dann ist die Entfernung λ der Senke s von der Quelle q in G beschränkt durch $\lambda \leq \frac{2|V|}{\sqrt{x^*_{sq}}} + 1$, wobei x^* OL ist. (OBdA $x^*_{sq} \geq 1$).*

Beweis. $V_i :=$ Knoten in Entfernung i von q, $i \leq \lambda$.

$$|V_i||V_{i+1}| \geq |\delta(V_o \cup \ldots \cup V_i)| \geq x^*(sq)$$

$\rightsquigarrow \max\{|V_i|, |V_{i+1}|\} \geq \sqrt{x^*_{sq}} = \alpha$. Es gibt also mindestens $\lfloor\frac{\lambda}{2}\rfloor$ Mengen (Schichten) mit $\geq \lceil\alpha\rceil$ Knoten $\rightsquigarrow \lfloor\frac{\lambda}{2}\rfloor\lceil\alpha\rceil \leq |V|$. □

Satz 2.13 $l \equiv 0, c \equiv 1$. *Mittels MPM findet man eine OL x^* in $O(|V|^{\frac{2}{3}}|E|)$ Operationen.*

Beweis. Aufwand pro Iteration $O(|E'|) \leq O(|E|)$. Anzahl der Iterationen sei k.

1.Fall $x^*_{sq} \leq |V|^{\frac{2}{3}} \rightsquigarrow k \leq |V|^{\frac{2}{3}}$.

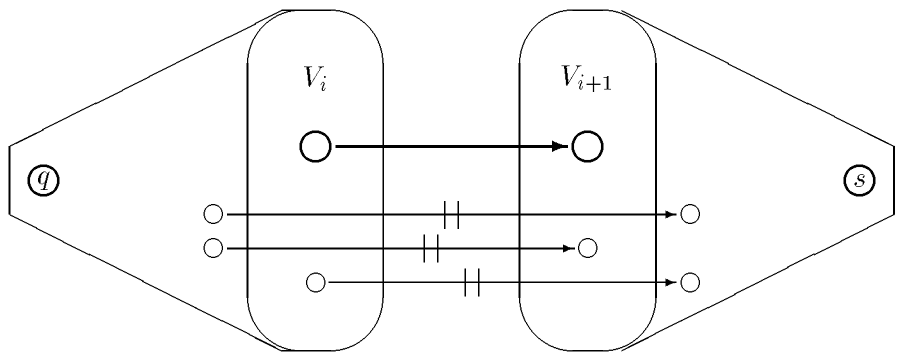


Abbildung II.22: Mögliche Kanten im Graphen

2.Fall Die k_1-te Iteration beginne mit $x_{sq} \leq x^*_{sq} - |V|^{\frac{2}{3}}$ und ende mit $x_{sq} > x^*_{sq} - |V|^{\frac{2}{3}}$. k_2 Anzahl des folgenden Iterationen $\leadsto k_2 \leq |V|^{\frac{2}{3}}$. λ Entfernung der Senke s von q in G_x. $\lambda \geq k_1$. Δ maximaler Wert eines Flusses in G_x auf sq. $\Delta = x^*_{sq} - x_{sq} \geq |V|^{\frac{2}{3}}$

$$\begin{aligned} k &= k_1 + k_2 < \lambda + |V|^{\frac{2}{3}} \leq \frac{2|V|}{\sqrt{\Delta}} + 1 + |V|^{\frac{2}{3}} \\ &\leq 3|V|^{\frac{2}{3}} + 1 \end{aligned}$$

Einfache 0-1 Flußprobleme

Spezialfall $c \equiv 1, l \equiv 0$ und $|\delta(v)| \leq 1 \vee |\delta(\bar{v})| \leq 1 \quad \forall\, v \in V$.

Lemma 2.14 *Im Fall einfacher $0-1$ Flußprobleme ist die Entfernung λ des Knotens s von q in G beschränkt durch $\lambda \leq \frac{|V|}{x^*_{sq}} + 1$.*

Beweis. Übungsaufgabe.

Satz 2.15 *Im Fall einfacher $0-1$ Flußprobleme bestimmt das MPM-Verfahren eine OL x^* mit $O(|V|^{\frac{1}{2}}|E|)$ Operationen.*

Bemerkung: Im Fall des Zuordnungsproblems (Heiratsproblems) ergibt sich ein $O(|V|^{\frac{5}{2}})$-Verfahren.

Beispiel:

#	Jahr	Entdecker	Komplexität
1	1969	Edmonds und Karp	$O(nm^2)$
2	1970	Dinic	$O(n^2m)$
3	1974	Karzanov	$O(n^3)$
4	1977	Cherkasky	$O(n^2\sqrt{m})$
5	1978	Malhotra, Kumar und Maheshwari	$O(n^3)$
6	1980	Galil	$O(n^{5/3}m^{2/3})$
7	1978	Shiloach	
	1980	Galil und Naamad	$O(nm\log^2 n)$
8	1982	Shiloach und Vishkin	$O(n^3)$
9	1983	Sleator und Tarjan	$O(nm\log n)$
10	1984	Tarjan	$O(n^3)$
11	1985	Gabow	$O(nm\log U)$
12	1985	Goldberg	$O(n^3)$
13	1986	Goldberg und Tarjan	$O(nm\log(\frac{n^2}{m}))$
14	1986	Bertsekas	$O(n^3)$
15	1987	Cheriyan und Maheshwari	$O(n^2\sqrt{m})$
16	1987	Ahuja und Orlin	$O(nm + n^2\log U)$
17	1988	Ahuja, Orlin und Tarjan	a) $O(nm + \frac{n^2\log U}{\log\log U})$
			b) $O(nm + n^2\sqrt{\log U})$
			c) $O\left(nm\log\left(\frac{n\sqrt{\log U}}{m} + 2\right)\right)$
18	1994	Tunçel	$O(n^2\sqrt{m})$

Quelle (Algorithmen bis 1988):
Ahuja, Magnanti, Orlin: Some Recent Advances in Network Flows
SIAM Review Vol. 33, No. 2, pp. 175–219, June 1991

3 Flüsse mit minimalen Kosten

Sei $a : E \to \boldsymbol{R}$ eine Kostenbewertung auf den Kanten von G, Kapazitäten $l \leq c$.

(3.1)
$$\min_{x\in P_C} \sum_{e\in E} a(e)x(e)$$

Inkrementkosten auf G_x

$$a_x(ij) := \begin{cases} +a(ij) & ij \in E_+ \\ -a(ji) & ij \in E_- \end{cases}$$

Lemma 3.1 *Sei Δx Fluß in G_x zu $x \in P_C$. Dann gilt:*

$$\sum_{e \in E} a(e)(x \oplus \Delta x)(e) = \sum_{e \in E} a(e)x(e) + \sum_{e \in E_x} a_x(e)\Delta x(e)$$

Kreise C in G_x

1. C heiß t negativ, falls $\sum_{e \in C} a_x(e) < 0$.

2. $0 \leq \mu \leq c_x(C) = \min_{e \in C} c_x(e)$. Inkrementfluß $\Delta x = \Delta x(C, \mu)$ auf C.

$$\Delta x(e) := \begin{cases} \mu & e \in C \\ 0 & \text{sonst} \end{cases} \quad \text{Speziell } \Delta x[C] := \Delta x(C, c_x(C)).$$

Satz 3.2 *Sei $x \in P_C$. Dann ist x OL zu (3.1) genau dann, wenn G_x keinen negativen Kreis enthält.*

Beweis. Lemma 3.1, Lemma 2.4: x ist genau dann OL, wenn G_x keinen Inkrementfluß Δx mit negativen Kosten enthält. (*)

Sei Δx beliebiger Inkrementfluß in G_x. Beh. $\Delta x = \sum_{i \in I} \Delta x(C_i, \mu_i)$ mit $|I| \leq |E|$. Konstruktion der Kreise: Sei $\Delta x(pq) > 0$. Flußerhaltung liefert Existenz eines Weges W von q nach p mit $\Delta x > 0$ auf W.

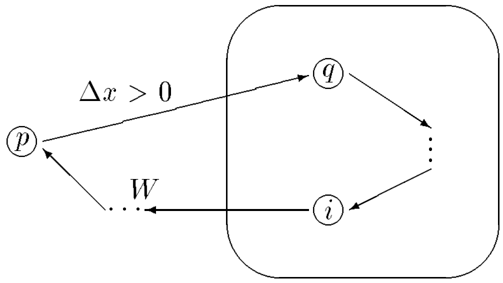


Abbildung II.23: Zur Konstruktion der Kreise

Abspalten: $\Delta x(W \cup pq, \mu)$ mit $\mu := \min_{e \in W \cup pq} \Delta x(e) \rightsquigarrow$ Reduziert in maximal $|E|$ Schritten den Ausgangsfluß auf 0.

$$\sum a_x(e)\Delta x(e) = \sum_{i \in I}((\sum_{e \in C_i} a_x(e))\mu_i)$$

d.h. (*) genau dann, wenn G_x keinen negativen Kreis enthält.

Algorithmus 3.1 (Verfahren negativer Kreise)

1. Bestimme $x \in P_C$.

2. Enthält G_x keinen negativen Kreis, stop [x optimal]
   Sei C ein negativer Kreis.

3. $x := x \oplus \Delta x[C]$ und gehe nach 2.

Bemerkung:

1. Negative Kreise mit Floyd $O(|V|^3)$.
2. a, l, c ganz. Startlösung $\bar{x}$ ganz. $\rightsquigarrow$ Folge von ganzen Flüssen mit streng antitonen Kosten. Sei x_* optimale Lösung $\rightsquigarrow$ Abschätzung: $O(\sum_{e\in E}|a(e)|\sum_{e\in E}|c(e)-l(e)||V|^3)$.

Minimum-Mean-Kreise

Sei C_x die Menge aller Kreise in G_x. Dann heißt $\tilde{C} \in C_x$ MM-Kreis, wenn

$$\frac{a_x(\tilde{C})}{|\tilde{C}|} = \min_{C\in C_x} \frac{a_x(C)}{|C|} =: \mu(x)$$

gilt.

MM-Kreis-Methode

neuer Schritt 2:
Bestimme einen MM-Kreis C in G_x. Falls $a_x(C) \geq 0$ stop [x optimal]

ϵ-optimale Flüsse

Sei $x \in P_C$. $\epsilon \geq 0$.

1. x heißt ϵ-optimal, falls eine Funktion $p : V \to \boldsymbol{R}$ ex. mit

$$a_{xp}(ij) := a_x(ij) + p(i) - p(j) \geq -\epsilon \quad \forall\, ij \in E_x$$

2. $\epsilon(x) := \min\{\epsilon \geq 0 | x\ \epsilon\text{-optimal}\}$
3. p heißt Potentialfunktion, falls $\epsilon(x) = 0$.

Bemerkung:

1. Sei $C \in C_x$. Dann gilt $a_{xp}(C) = a_x(C)$.
2. $\epsilon(x) = 0 \to G_x$ enthält keine negativen Kreise $\leftrightarrow x$ optimal.

Satz 3.3 *Sei $x \in P_C$. Dann ist x optimal genau dann, wenn x 0-optimal.*

Beweis. "$\to$": Sei x optimal $\rightsquigarrow G_x$ enthält keine negativen Kreise. OBdA G_x stark zusammenhängend. Wähle $s \in V$ beliebig. Berechne kürzeste Wege von s zu allen Knoten $i \neq s$. Die Länge eines kürzesten Weges bzgl. a_x sei $p(i)$ für Knoten i, $p(s) = 0$. Dann gilt: $p(j) \leq a_x(ij) + p(i) \quad \forall\, ij \in E_x$. D.h. x ist 0-optimal. □

Lemma 3.4 *Sei a ganz, $x \in P_C$ mit $\epsilon(x) < \frac{1}{|V|}$. Dann ist x OL zu (3.1).*

Beweis. Sei $p : V \to \boldsymbol{R}$ mit $a_{xp} \geq -\epsilon(x)$. Sei $C \in C_x$

$$a_x(C) = a_{xp}(C) \geq -|C|\epsilon(x) > -\frac{|C|}{|V|}. \tag{3.2}$$

Für einfache Kreise $C \leadsto a_x(C) > -1 \overset{a \text{ ganz}}{\leadsto} a_x(C) \geq 0$. Da jeder Kreis Summe einfacher Kreise, ist x OL. □

Lemma 3.5 *Sei $x \in P_C, \epsilon(x) > 0$. Dann gilt: $\mu(x) = -\epsilon(x)$.*

Beweis. Wegen (3.2) gilt $\frac{a_x(C)}{|C|} \geq -\epsilon(x) \quad \forall\, C \in C_x$ also $\mu \geq -\epsilon(x)$. Andererseits sei $a_x^\mu := a_x - \mu(x) \leadsto a_x^\mu(C) = a_x(C) - |C|\mu(x) \geq 0 \quad \forall\, C \in C_x$. Sei $p : V \to \boldsymbol{R}$ Potentialfunktion zu a_x^μ, d.h. $a_{xp}^\mu \geq 0 \leadsto a_{xp} \geq \mu(x)$ d.h. x ist $-\mu(x)$-optimal $\leadsto \epsilon(x) \leq -\mu(x)$. Also insgesamt $\epsilon(x) = -\mu(x)$. □

Lemma 3.6 *Seien $x \in P_C$ und C MM-kreis in G_x. Dann gilt $\epsilon(x + \Delta x[C]) \leq \epsilon(x)$.*

Beweis. Sei $l = |C|$. $p : V \to \boldsymbol{R}$ mit $a_{xp} \geq -\epsilon(x) \leadsto a_{xp}(C) = a_x(C) = l\mu(x) = -l\epsilon(x) \leadsto a_{xp}(e) = -\epsilon(x) \quad \forall\, e \in C$. Sei $ij \in E_{x \oplus \Delta x}$. Falls $ij \in E_x$:

$$a_{x \oplus \Delta x\ p}(ij) = a_{xp}(ij) \geq -\epsilon(x)$$

Falls $ij \notin E_x \leadsto ji \in C$:

$$a_{x \oplus \Delta x\ p}(ij) = -a_x(ji) + p(i) - p(j) = -(a_x(ji) + p(j) - p(i)) = -a_{xp}(ji) = \epsilon(x) > 0$$

$\leadsto x \oplus \Delta x$ ist $\epsilon(x)$-optimal $\leadsto \epsilon(x \oplus \Delta x) \leq \epsilon(x)$. □

Lemma 3.7 *$|V| = n, |E| = m$. Sei $x_0 \in P_C$. Seinen C_i MM-Kreise zu x_{i-1}. $x_i := x_{i-1} \oplus \Delta x[C_i]$ für $i = 1, \ldots, m$. Dann gilt $\epsilon(x_m) \leq (1 - \frac{1}{n})\epsilon(x_0)$.*

Beweis. Sei $p : V \to \boldsymbol{R}$ mit $a_{x_0 p} \geq -\epsilon(x_0)$. $N_k := \{ij \in E_{x_k} | a_{x_k p}(ij) < 0\}$.

1.Fall $a_{x_k p}(ij) < 0$ für alle $ij \in C_k, \quad k = 1, \ldots, m$. Entwicklung von E_{x_k} zu E_{k+1}.

1. $ij \in E_{x_{k+1}} \setminus E_{x_k} \leadsto ji \in C_k \leadsto a_{x_{k+1}p}(ij) = \epsilon(x_k) = -a_{x_k}p(ji) > 0$ also $N_{k+1} \subset N_k$.
2. Andererseits: $\Delta x[C_k](e) = c_{x_{k-1}}(e)$ für eine Kante $e \in C_k \subseteq N_{k-1} \leadsto e \notin N_{k+1} \leadsto N_m = \emptyset \leadsto \epsilon(x_m) = 0$.

2.Fall Sei k minimal mit $a_{x_{k-1}p}(ij) \geq 0$ für ein $ij \in C_k, \quad 1 \leq k \leq m$

$$\epsilon(x_m) \leq \epsilon(x_{k-1}) = \frac{-a_{x_{k-1}}(C_k)}{|C_k|} \leq \frac{(|C_k| - 1)\epsilon(x_0)}{|C_k|} \overset{C_k \text{ einfach}}{\leq} \frac{n-1}{n}\epsilon(x_0)$$

□

Satz 3.8 *Sei $G = (V, E)$ mit $|V| = n, |E| = m$. Sei a ganz, $a_{\max} := \max_{e\in E} |a(e)|$. Dann terminiert die MM-Methode nach $O(mn \log(na_{\max}))$ Iterationen.*

Beweis. Sei $x_0 \in P_C \overset{p\equiv 0}{\leadsto} \epsilon(x_0) \leq a_{\max}$. Nach k Iterationen

$$\epsilon(x_k) \leq (1 - \frac{1}{n})^{\lfloor \frac{k}{m} \rfloor} \epsilon(x_0)$$

Für $k = mn\lceil \log(na_{\max})\rceil$ gilt:

$$\begin{aligned} \epsilon(x_k) &= (1 - \frac{1}{n})^{n \log(na_{\max})} a_{\max} \\ &< e^{-\log(na_{\max})} a_{\max} = \frac{1}{n} \end{aligned}$$

$\leadsto x_k$ optimal. □

Bemerkung:

1. MM -Kreise kann man in $O(nm)$ bestimmen (Übung) $\leadsto O(n^2m^2 \log(na_{\max}))$
2. $O(mn^2L))$ mit $L = O(a_{\max})$ für Verfahren mit Minimum-Ratio-Kreisen. Wallacher, Zimmermann, 1992.

Satz 3.9 *Sei $x \in P_C$. Dann ist x genau dann optimal, wenn ein $u : V \to \boldsymbol{R}$ mit*

$$a(ij) + u(i) - u(j) \begin{cases} \geq 0 & \textit{falls } l(e) = x(e) \\ = 0 & \textit{falls } l(e) < x(e) < c(e) \\ \leq 0 & \textit{falls } c(e) = x(e) \end{cases}$$

für alle $ij = e \in E$ mit $l(e) < c(e)$ existieren.

Beweis. Äquivalent zu: u ist ein Potential.

1. $l(e) = x(e): \quad e \in E_x, \quad a_{xu}(ij) \geq 0$
2. $l(e) < x(e) < c(e): \quad ij, ji \in E_x, \quad a_{xu}(ij) = a_{xu}(ji) = 0$
3. $x(e) = c(e): \quad ji \in E_x, \quad a_{xu}(ji) \geq 0$ □

Kilterzahlen

Sei x Fluß. $u : V \to \boldsymbol{R}$. Für $e \in E$ ist $K_e(x, u) \hat{=}$ waagerechter Abstand zur Kilterlinie. $K(x, u) := \sum_{e\in E} K_e(x, u)$. Genau für $K(x, u) = 0$ ist x optimal.

Idee: systematisch Kilterzahlen reduzieren.

Wahl von $e \in E$ mit $K_e(x, u) > 0$

$e = rs$. $x(e) \leadsto x(e) \pm \delta$ durch $\Delta x(pq) = \delta$ mit $pq \in \{rs, sr\}$, wobei, falls $rs, sr \in E_x$, pq so gewählt wird, daß $a_{xu}(pq) \leq 0$ gilt.

Idee: Zur Flußänderung konstruiere einen Kreis C in G_x, so daß $x' = x \oplus \Delta x(C, \delta)$ "besser" wird.

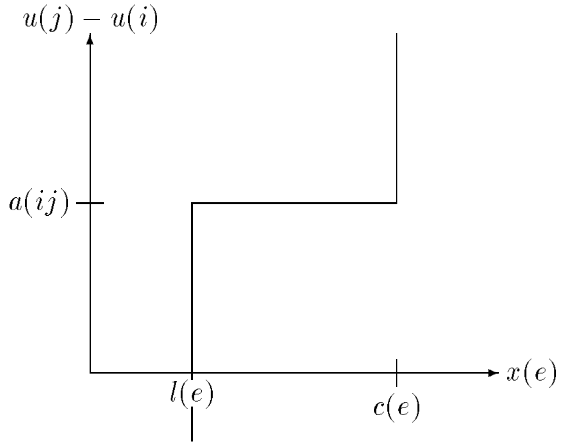


Abbildung II.24: Die Kilterlinie

Neue Kapazität auf G_x

$$c_{xu}(ij) := \begin{cases} l(e) - x(e) & \text{falls } l(e) > x(e), a_{xu}(e) > 0 \\ x(ji) - c(ji) & \text{falls } c(ji) < x(ji), a_{xu}(e) < 0 \\ c_x(ij) & \text{andernfalls} \end{cases}$$

verhindert zu großes δ.

Lemma 3.10 *Sei $x' = x \oplus \Delta x$ mit $\Delta x \leq c_{xu}$. Gilt für alle $ij \in E_x$ mit $l(ij) \leq x(ij)$ bzw. $x(ji) \leq c(ji)$*

$$a_{xu}(ij) > 0 \Rightarrow \Delta x(ij) = 0,$$

so gilt $K(x', u) \leq K(x, u)$.

Beweis. Siehe Abbildung II.25. □

Änderung der Duallösung

$u' := u + \lambda$ mit $K_{ij}(x, u') \leq K_{ij}(x, u)$.

Lemma 3.11 *Falls gilt:*

$$\min(0, a_{xu}(ij)) \leq \min(0, a_{xu'}(ij)\forall ij \in E_x,$$

so ist $K(x, u') \leq K(x, u)$.

Beweis. Siehe Abbildung II.26. □

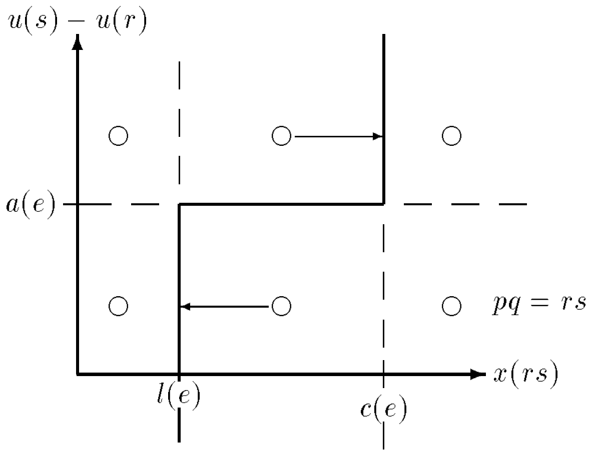


Abbildung II.25: Mögliche Lage der Kante pq im Kilterdiagramm

Kürzeste Wege von q aus in G_x

$S = \{v | v$ erreichbar von q in $G_x\}$. $d : E_x \to \boldsymbol{R}_+$ mit $d(ij) := \max(0, a_{xu}(ij))$ $(ij \in E_x)$.

$\lambda : V \to \boldsymbol{R}_+$ mit $\lambda(j) := \begin{cases} \text{Entfernung von } j \text{ bzgl. } d & j \in S \\ \lambda_\infty & j \notin S \end{cases}$

mit $\lambda_\infty := \max_{ij \in E_x, i \notin S, j \in S}(\lambda(j) - a_{xu}(ij))$

$\lambda_\infty := 0$, falls keine solche Kante.

Algorithmus 3.2 ("Out-of-kilter"-Algorithmus)

1. Setze $u := 0$.

   Finde einen Startfluß x, der die Flußerhaltung erfüllt.

2. Bestimme den Inkrementgraphen G_x und die Kilterzahlen.

3. Falls "out-of-kilter"-Kanten existieren, wähle eine "out-of-Kilter"-Kante pq in G_x.

   Anderenfalls stop (x optimal).

4. Bestimme $\lambda : V \to \boldsymbol{R}, S \subseteq V$.

5. Falls $(x(ij) < l(ij) \vee c(ji) < x(ji))$ für ein $ij \in \delta_x(\bar{S})$ dann stop. $(P_C = \emptyset)$

6. $u := u + \lambda$.

7. Falls $p \in S$ und $K_r s(x, u) > 0$ dann $x := x \oplus \Delta x(C, \delta)$.

8. Falls $K_{rs}(x, u) = 0$, gehe nach 2,

   anderenfalls gehe nach 3.

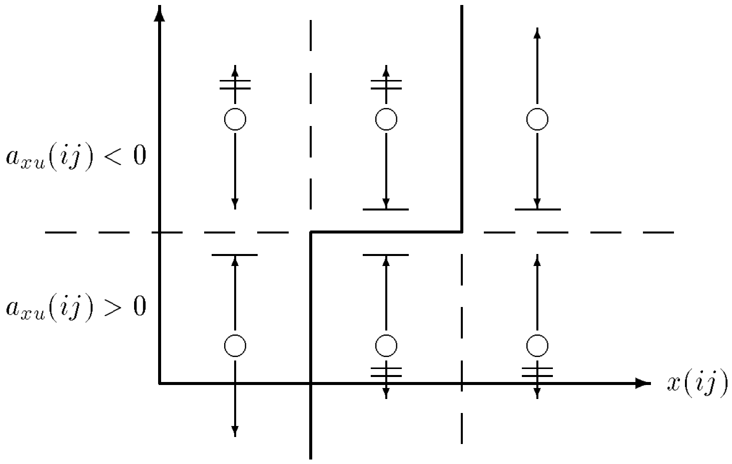


Abbildung II.26: Änderung des Kilterdiagramms bei Änderung der Dualvariablen

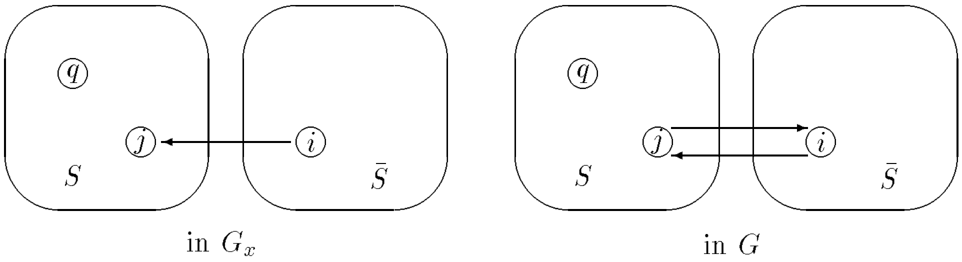


Abbildung II.27: Kanten in G_x und in G

Lemma 3.12 *Gibts es eine Kante $ij \in \delta_x(\bar{S})$ mit $x(ij) < l(ij)$ oder $c(ji) < x(ji)$, so ist $P_c = \emptyset$.*

Beweis. Nach Konstruktion $\delta_x(S) = \emptyset$.

1. $ij \in E_+ \overset{ji \notin E_x}{\leadsto} x(ij) \leq l(ij)$
2. $ij \in E_- \overset{ji \notin E_x}{\leadsto} c(ji) \leq x(ji)$

$\leadsto c(\delta(S)) \leq x(\delta(S)) = x(\delta(\bar{S})) \leq l(\delta(\bar{S})) \leadsto c(\delta(S)) < l(\delta(\bar{S})) \overset{2.7}{\leadsto} P_c = \emptyset.$ □

Bemerkung: Falls $P_c \neq \emptyset$, so gilt $x(ij) = l(ij)$ oder $c(ji) = x(ji)$ für alle $ij \in \delta_x(S)$.

Lemma 3.13 *Sei $u' := u + \lambda$. Dann gilt $K(x, u') \leq K(x, u)$. Ist die Voraussetzung von Lemma 3.12 nicht erfüllt und gilt:*

1. $p \in \bar{S}$ oder

2. $W = qp$ ist ein KW von q nach p in G_x bzgl. d,

so ist $K_{rs}(x, u') = 0$ und daher $K(x, u') < K(x, u)$.

Beweis. Sei $ij \in E_x$. Zu zeigen: Bedingung in Lemma 3.11.

1. $ij \in \delta_x(\bar{S}) \leadsto \lambda(i) = \lambda_\infty \geq \lambda(j) - a_{xu}(ij) \leadsto a_{xu'}(ij) = a_{xu}(ij) + \lambda(i) - \lambda(j) \geq 0 \leadsto$ Bedingung erfüllt.

   Wegen Voraussetzung: $x(ij) = l(ij)$ oder $x(ji) = c(ji) \leadsto K_{ij}(x, u') = 0$ oder $K_{ji}(x, u') = 0$, also insbesondere $K_{rs}(x, u') = 0$ falls $p \notin \bar{S}$.

2. $i, j \in \bar{S}$. $\lambda(i) = \lambda(j) = \lambda_\infty \leadsto a_{xu}(ij) = a_{xu'}(ij) \leadsto$ Bedingung erfüllt.

3. $i, j \in S$. Nach Konstruktion von λ gilt: $\lambda(j) \leq \lambda(i) + d(ij)$ (längs eines kürzesten Weges gilt "=") $\leadsto a_{xu'}(ij) = a_{xu}(ij) + \lambda(i) - \lambda(j) \geq a_{xu}(ij) - \max(0, a_{xu}(ij))$.

$$\begin{array}{ll} \text{1.Fall} & a_{xu}(ij) < 0 \leadsto a_{xu'}(ij) \geq a_{xu}(ij) \\ \text{2.Fall} & a_{xu}(ij) \geq 0 \leadsto a_{xu'}(ij) \geq 0 \end{array} \tag{3.3}$$

   In jedem Fall ist die Bedingung erfüllt.

Sei nun $W = qp$ ein KW von q nach p bzgl. d. Nach Wahl von pq gilt nun

$$a_{xu}(qp) > 0 \leadsto a_{xu'}(qp) = 0 \leadsto a_{xu'}(pq) = 0 \leadsto K_{rs}(x, u') = 0$$

□

Bemerkung: Längs kürzester Wege in S gilt (3.3) mit "=". Längs kürzester Wege in S gilt $a_{xu'}(ij) \leq 0$ für alle $ij \in KW$.

Neuer Fluß x'

Sei die Bedingung in Lemma 3.12 nicht erfüllt, $p \in S$, $K_{rs}(x, u') > 0$. Sei W ein $KW_{q \leadsto p}$ in G_x bzgl. d. Dann setze $C := W \cup pq$. $\delta := \min_{e \in C} c_{x,u'}(e)$ und $x' := x \oplus \Delta x(C, \delta)$.

Lemma 3.14 *Dann gilt $K(x', u') < K(x, u')$.*

Beweis. Sei $ij \in W$. Dann gilt $a_{xu'}(ij) < 0$ (Bemerkung). Sei $qp \in E_x$: $a_{xu}(pq) > 0 \leadsto a_{xu'}(qp) \geq 0 \leadsto a_{xu'}(pq) \leq 0$.

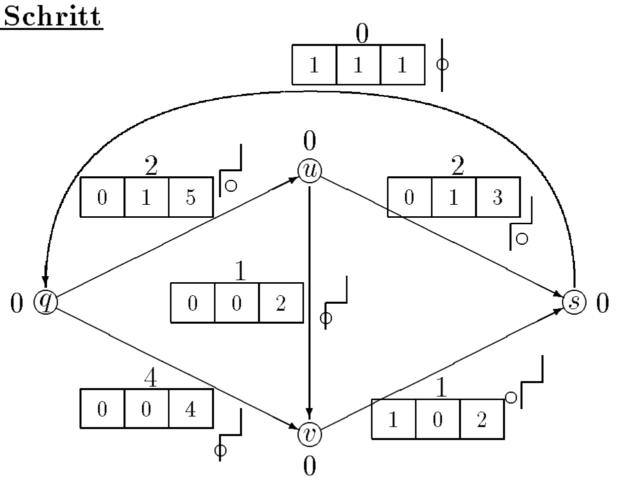
Anderenfalls ($qp \notin E_x$)

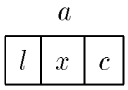
$$\left.\begin{array}{l} l(pq) \leq x(pq) \overset{qp \notin E_x}{\leadsto} l(pq) = x(pq) \\ x(qp) \leq c(qp) \overset{qp \notin E_x}{\leadsto} x(qp) = c(qp) \end{array}\right\} \overset{K_{rs}(x,u')>0}{\leadsto} a_{xu'}(pq) \leq 0$$

Wegen $\delta > 0$, folgt $K_{rs}(x', u') < K_{rs}(x, u') \leadsto K(x', u') < K(x, u')$. □

Beispiel:

1. Schritt



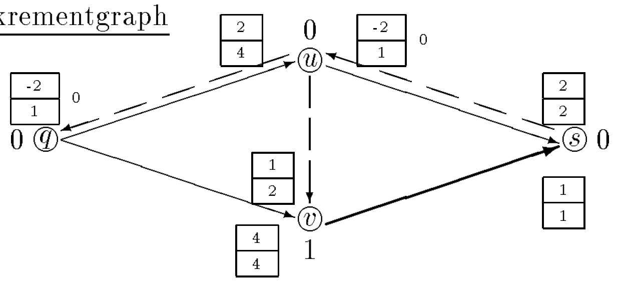


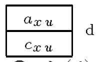
i $u(i)$

wobei gilt:

$vs \equiv rs \equiv pq$

Inkrementgraph





i $\lambda(i)$

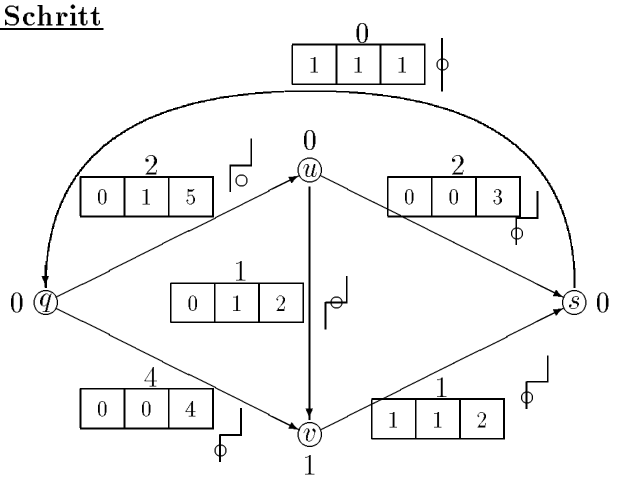
$a_{xu}(ij) = a_x(ij) + u_i - u_j$

also: $v \equiv p \in S$

$\rightsquigarrow$ Kreis $C \equiv (s\,u\,v)$

$c_{xu'}(C) = 1 = \delta$

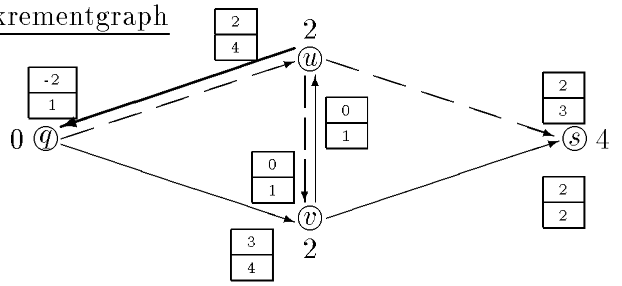
2. Schritt



$x = x \oplus \Delta x(C, \delta)$

$u = u + \lambda$

Inkrementgraph



also:

$u \equiv p \in S$,

$w = (qu) \equiv (qp)$

$\rightsquigarrow K_{qu}(x, u + \lambda) = 0$

Komplexität der “out of kilter”-Methode

Sukzessive Reduktion der Kilterzahlen mittels des Reduktionsalgorithmuses.

1. Start mit "geeigneten" Daten: l, c, x^0, u^0, a ganz $\rightsquigarrow$ x stets ganz im Algorithmus $\rightsquigarrow O(K(x^0, u^0)|V|^2)$.
2. $0 \leq l \leq c \rightsquigarrow K(0,0) \leq c(E)$.
3. $0-1$-Flußprobleme ($l \leq c$ $0-1$-Vektoren) $\rightsquigarrow K(0,0) \leq E \rightsquigarrow O(|E||V|^2)$.

Skalierung von c

$p := \lceil \log_2(\max_e c(e)) \rceil$. Betrachte eine Folge von Problemen für $k = 0, 1, \ldots, p$ mit

$$l^k(e) := \left\lfloor \frac{l(e)}{2^{p-k}} \right\rfloor \quad c^k(e) := \left\lceil \frac{c(e)}{2^{p-k}} \right\rceil$$

Das zugehörige Polyeder sei P_C^k.

Bemerkung:

1. $P_C \neq \emptyset \rightsquigarrow P_C^k \neq \emptyset \quad \forall\, k$, denn $2^{p-k} l^k \leq l \leq c \leq 2^{p-k} c^k$.
2. $l^{k+1} - 2l^k \leq 1$ und $2c^k - c^{k+1} \leq 1 \quad \forall\, k$, denn $\lfloor 2\alpha \rfloor - 2\lfloor \alpha \rfloor \leq 1$ und $2\lceil \alpha \rceil - \lceil 2\alpha \rceil \leq 1$.

Startschritt

$x \equiv 0, u \equiv 0$ Startlösung $c^0(e), l^0(e) \in \{0,1\} \rightsquigarrow$ in $O(mn^2)$ berechnet "out of kilter" die Lösung. $m = |E|, n = |V|$.

Iterationsschritt $k \to k+1$

$x^{k+1} := 2x^k, u^{k+1} := u^k \rightsquigarrow K_e(x^{k+1}, u^{k+1}) \in \{0,1\} \rightsquigarrow$ in $O(mn^2)$ wird die optimale Lösung bestimmt.

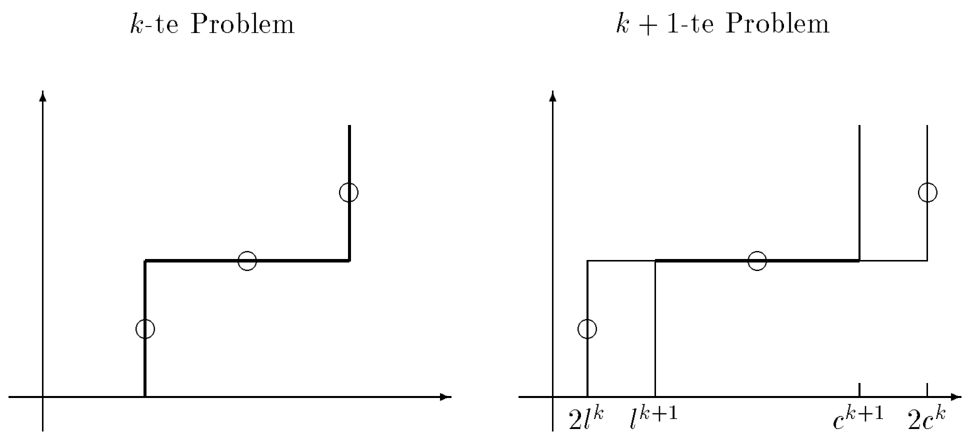


Abbildung II.28: Übergang von Problem k zu Problem $k+1$

d.h. insgesamt erhält man die Komplexität: $O(pmn^2)$ für alle $k = 0, 1, \ldots, p$ Iterationen.

Satz 3.15 *Bricht das Verfahren in einer Iteration nicht ab, so erhöht sich keine der Kilterzahlen und K_{rs} wird echt reduziert.*

Nichtendliche Kapazitäten

Falls l und c teilweise nichtendlich sind, überprüfe, ob es negative Kreise in $\tilde{E} := \{ij \in E_x | \bar{c}(ij) = a\}$ gibt. Wenn ja, so ist das Verfahren nicht endlich.

Beispiel:

Polynomiale Algorithmen für das Minimum-Cost-Flow-Problem			
#	Jahr	Entdecker	Komplexität
1	1972	Edmonds und Karp	$O(m \log U S(n,m,C))$
2	1980	Röck	$O(m \log U S(n,m,C))$
3	1980	Röck	$O(n \log C M(n,m,U))$
4	1985	Bland und Jensen	$O(n \log C M(n,m,U))$
5	1987	Goldberg und Tarjan	$O(nm \log(n^2/m) \log(nC))$
6	1988	Goldberg und Tarjan	$O(nm \log n \log(nC))$
7	1988	Ahuja et al.	$O(nm \log\log U \log(nC))$

Stark polynomiale Algorithmen für das Minimum-Cost-Flow-Problem			
#	Jahr	Entdecker	Komplexität
1	1985	Tardos	$O(m^4)$
2	1984	Orlin	$O(m^2 \log n S(n,m))$
3	1986	Fujishige	$O(n^2 \log n S(n,m))$
4	1986	Galil und Tardos	$O(n^2 \log n S(n,m))$
5	1987	Goldberg und Tarjan	$O(nm^2 \log n \log(n^2/m))$
6	1988	Goldberg und Tarjan	$O(nm^2 \log^2 n)$
7	1988	Orlin	$O(m \log n \log S(n,m))$

Polynomiale Algorithmen		
Jahr	Entdecker	Komplexität
1982	Johnson	$S(n,m,C) = O(m \log\log C)$
1988	Ahuja, Mehlhorn, Orlin und Tarjan	$S(n,m,C) = O(m + n\sqrt{\log C})$
1987	Ahuja, Orlin und Tarjan	$M(n,m,C) = O(nm \log \left(\frac{n\sqrt{\log U}}{m} + 2\right))$

Stark polynomiale Algorithmen		
Jahr	Entdecker	Komplexität
1984	Fredman und Tarjan	$S(n,m) = O(m + n \log n)$
1986	Goldberg und Tarjan	$M(n,m) = O(nm \log \left(\frac{n^2}{m}\right))$
1987	Cheriyan und Maheshwari	$M(n,m) = O(n^2\sqrt{m})$

Notation:

C = Maximale Kosten
U = Maximale Kantenkapazität
$S()$ = Komplexität für einen kürzeste-Wege-Algorithmus mit nichtnegativen Längen
$M()$ = Komplexität für einen Maxflow-Algorithmus

Quelle:

Ahuja, Magnanti, Orlin: Some Recent Advances in Network Flows
SIAM Review Vol. 33, No. 2, pp.175–219, June 1991

4 Anwendungen: Projektplanung, Transportproblem

4.1 Projektplanung

$G = (V, E)$ azyklisch.

Teilprojekte $ij = e$	Kante e
e vor e'	es exisitiert ein gerichteter Weg von j nach μ
$b(e)$	minimale Bearbeitungsdauer
$a(e)$	normale Bearbeitungsdauer
$t(e)$	Bearbeitungsdauer bei Kosten $c(e)(a(e) - t(e)), b(e) \leq t(e) \leq a(e)$
$u(i)$	Knotenvariable $\hat{=}$ Zeitpunkt, zu dem alle Teilprojekte vor Knoten i abgeschlossen sinc
$u(q), u(s)$	Start/Endzeitpunkt
T	maximal erlaubte Projektdauer

Problem

$$
\begin{array}{lrcllll}
C := \min & \sum_e c(e)(a(e) - t(e)) & \rightarrow & -C & = & -\sum c(e)a(e) + \max \sum c(e)t(e) & \\
\text{unter} & u(s) - u(q) & \leq & T & \leftarrow & x(sq) & \\
(4.1) & u(i) - u(j) + t(e) & \leq & 0 & \leftarrow & x(e) & \forall\, e = ij \\
 & b(e) & \leq & t(e) & \leftarrow & \beta(e) & \forall\, e = ij \\
 & t(e) & \leq & a(e) & \leftarrow & \alpha(e) & \forall\, e = ij
\end{array}
$$

Das duale Problem

$$
(4.2)\qquad \begin{array}{lrcl}
 & -\sum a(e)c(e) & + & \min[Tx(sq)+\sum a(e)\alpha(e)-\sum b(e)\beta(e)] \\
\text{unter} & \partial x(i) & = & 0 \\
 & x(e)+\alpha(e)-\beta(e) & = & c(e) \\
 & x,\alpha,\beta & \geq & 0
\end{array}
$$

Komplementaritätsbedingungen

$$
\begin{array}{rcl}
x(e)>0 & \rightsquigarrow & t(e)=u(j)-u(i) \\
x(sq)>0 & \rightsquigarrow & T=u(s)-u(q) \\
\alpha(e)>0 & \rightsquigarrow & t(e)=a(e) \\
\beta(e)>0 & \rightsquigarrow & t(e)=b(e) \\
u(q)=0 & \text{wählbar} &
\end{array}
$$

Duale Zielfunktion

$$
z = Tx(sq)+\sum(\underbrace{a(e)-b(e)}_{\geq 0})\beta(e)-\sum a(e)x(e)
$$

da $\beta(e)-x(e)=\alpha(e)-c(e)$. In Abhängigkeit von $x(e)$ kann man $\beta(e)$ optimal wählen:

$$
(4.3)\qquad \begin{array}{lll}
\beta(e)=0 & \text{falls} & x(e)\leq c(e) \\
\beta(e)=x(e)-c(e) & \text{falls} & x(e)>c(e)
\end{array}
$$

$$
z = Tx(sq)+\sum T(e)(x(e))
$$

mit

$$
T(e)(x(e)) := \begin{cases} -a(e)x(e) & x(e)\leq c(e) \\ -a(e)c(e)-b(e)(x(e)-c(e)) & x(e)>c(e) \end{cases}
$$

Optimale Lösung von (4.1)

$$
(4.4)\qquad \begin{array}{rcl}
u(i) & := & -(\hat{u}(i)-\hat{u}(q)) \text{ Normierung auf } u(q)=0. \\
t(e) & := & \begin{cases} a(e) & x(e)<c(e) \\ u(j)-u(i) & x(e)=c(e) \\ b(e) & x(e)>c(e) \end{cases}
\end{array}
\qquad
\begin{array}{c|c}
\beta(e) & \alpha(e) \\ \hline
0 & c(e)-x(e) \\
0 & 0 \\
x(e)-c(e) & 0
\end{array}
$$

d.h. die Komplementaritätsbedingungen für α,β sind erfüllt.

Satz 4.1 *Sind die Kilterbedingungen für einen Fluß x erfüllt, so definieren (4.3) und (4.4) komplementäre zulässige Lösungen $(x(sq),x,\alpha,\beta)$ und (u,t) von (4.1) und (4.2).*

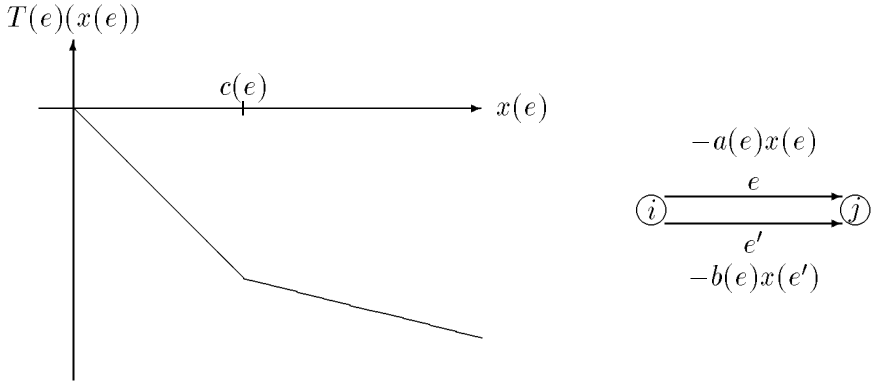


Abbildung II.29: Die Funktion $T(e)(x(e))$

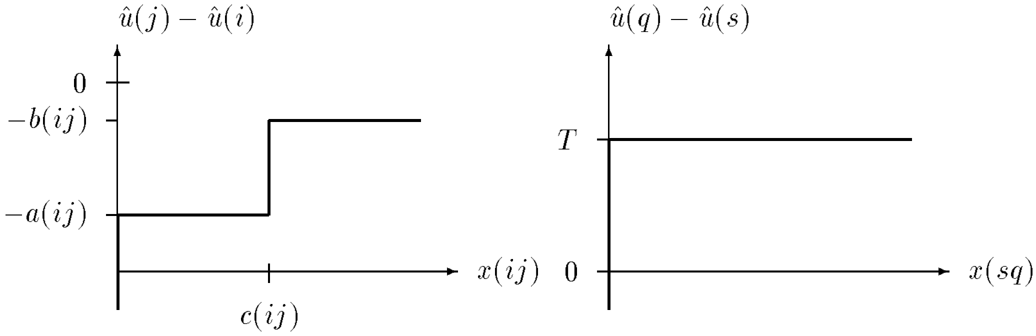


Abbildung II.30: Kilterlinien bei dualem Problem

Beweis. Zulässigkeit trivial nach Kilterdiagramm, bzw. (4.4).

$$\begin{array}{lcl}
x(sq) > 0 & \leadsto & u(s) = -(\hat{u}(s) - \hat{u}(q)) = T \\
 & \leadsto & \text{Komplementaritätsbedingungen für } x(sq) \text{ erfüllt.} \\
0 < x(e) < c(e) & \overset{(4.4)}{\leadsto} & t(e) = a(e) = u(j) - u(i) \\
 & \leadsto & \text{Komplementaritätsbedingungen für } x(e) \text{ erfüllt.} \\
0 < x(e) = c(e) & \overset{(4.4)}{\leadsto} & t(e) = u(j) - u(i) \\
 & \leadsto & \text{Komplementaritätsbedingungen für } x(e) \text{ erfüllt.} \\
c(e) < x(e) & \overset{(4.4)}{\leadsto} & t(e) = b(e) = u(j) - u(i) \\
 & \leadsto & \text{Komplementaritätsbedingungen für } x(e) \text{ erfüllt.}
\end{array}$$

□

Parametrische Analyse in Abhängigkeit von T

1. T hinreichend groß $\overset{\min}{\leadsto}$ $x(sq) = 0 \leadsto x \equiv 0 \leadsto C = 0$. Genauer: $T \geq T_1 \leadsto C = 0$, T_1 ist Gewicht eines längsten Weges bzgl. $a(e)$ in $O(n^2)$ bestimmbar, da der Graph

$G = (V, E)$ azyklisch ist.

2. T sukzessive absenken. Startlösung: OL für T, genau eine Kante, nämlich sq ist anfangs nicht "im Kilter".

3. T hinreichend klein $\rightsquigarrow$ negativer Kreis mit unbeschränkten Kapazitäten entsteht $\rightsquigarrow$ (4.1) unlösbar. $T < T_2 \hat{=}$ Gewicht eines längsten Weges bzgl. $b(e)$ (in $O(n^2)$).

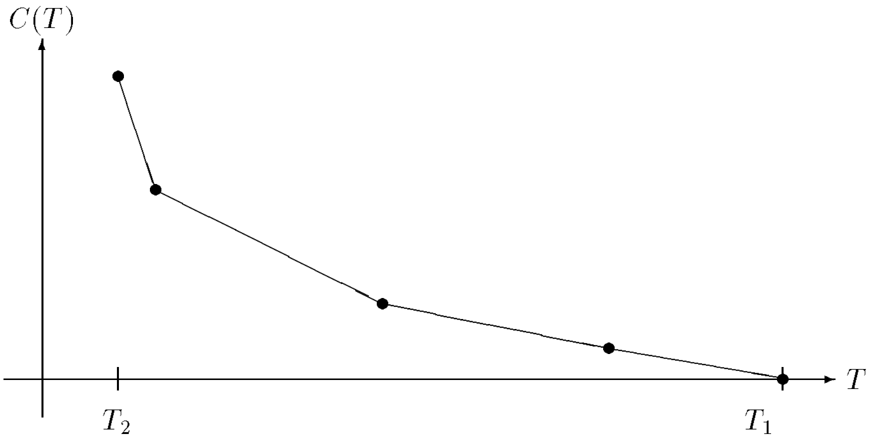


Abbildung II.31: parametrische Analyse in Abhängigkeit von T

4.2 Transportproblem als Flußproblem

Flußprobleme sind ebenfalls auf Transportprobleme reduzierbar (s. Lawler S. 191)

4.3 Kilterbedingungen beim Transportproblem

$x \equiv 0, u \equiv 0 \rightsquigarrow a_i, b_j$ ganz $\rightsquigarrow O(\sum a_i n^2)$ Verfahren. Zuordnungsprobleme: $a \equiv 1, b \equiv 1 \rightsquigarrow O(n^3)$ Verfahren.

5 Vollständige Unimodularität

Polyeder P

$P = \{x \in \boldsymbol{Q}^n | Ax \leq b\}$ mit rationalen Daten.

Seiten von P

Sei $A'x \leq b'$ Teilsystem von $Ax \leq b$. Dann heißt $\emptyset \neq S := \{x \in P | A'x = b'\}$ eine Seite von P. ($\dim S = 0$ Ecke, $\dim S = \dim P - 1$ Facette)

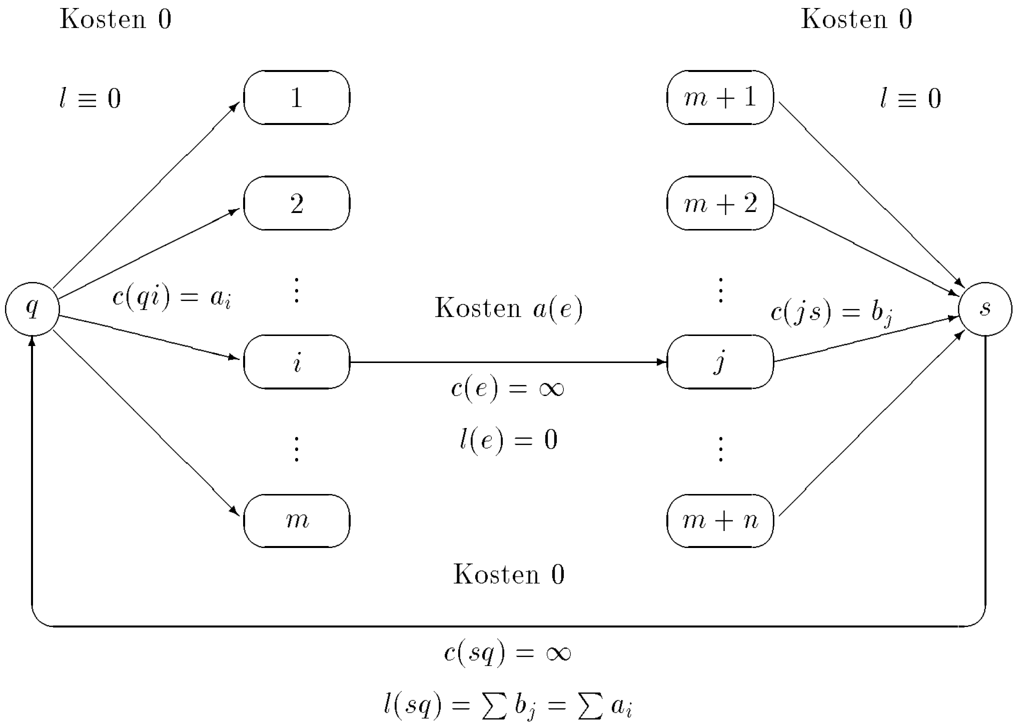


Abbildung II.32: erweiterter Transportgraph

Ganze Polyeder

Jede (minimale) Seite von P enthält einen ganzzahligen Punkt.
(Ist ein spitzes Polyeder ganz, so ist also jede Ecke ganz.)

Bemerkung: LP über ganzen Polyedern sind einfach ganzzahlig lösbar, da insbesondere die Seite der Optimallösungen einen ganzzahligen Punkt enthält.

Unimodularität

A ganz, $m \times n$

1. $m = n$: $\det A \in \{\pm 1\}$
2. $m = \mathrm{rg} A$: jede reguläre $m \times m$ Teilmatrix B erfüllt $\det B \in \{\pm 1\}$

Vollständige Unimodularität

Jede quadratische Teilmatrix B erfüllt $\det B \in \{0, \pm 1\}$. A vollst. unimodular $\rightsquigarrow$ A ist $(0, \pm 1)$-Matrix.

Satz 5.1 *A vollständig unimodular, b ganz. Dann ist $P = \{x | Ax \leq b\}$ ganz.*

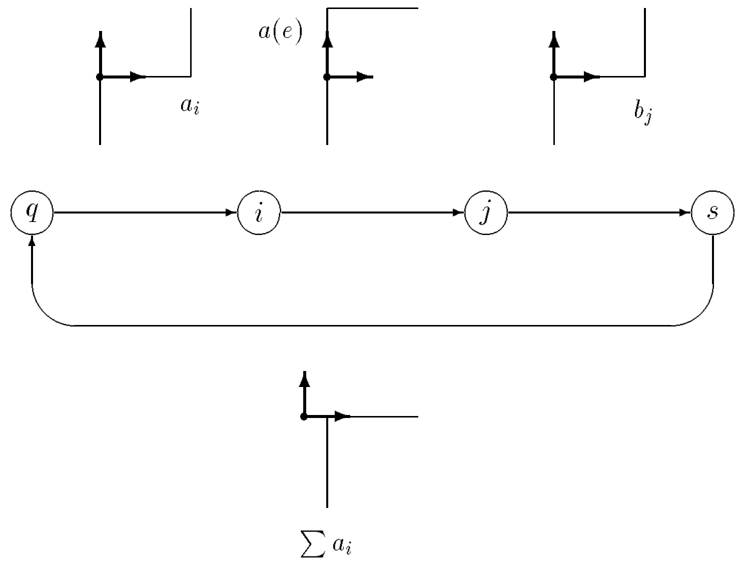


Abbildung II.33: Kilterbedingungen der Kanten im Transportgraph

Beweis. $S = \{x \in P | A'x = b'\}$ sei minimale Seite von P. A' habe vollen Zeilenrang. OBdA $A' = [U|V]$ mit $\det U \in \{\pm 1\} \rightsquigarrow x := [U^{-1}b'|0]^T \in S$ ganz.

Bemerkung:

1. A vollständig unimodular $\leftrightarrow [E| - A| - A^T]$ vollständig unimodular.

2. A vollständig unimodular $\leftrightarrow [A|E]$ unimodular.

Korollar 5.2 *A vollständig unimodular, b, c ganz. Dann gilt der ganzzahlige Dualitätssatz:*

$$\max\{c^T x | Ax \leq b, x \in \boldsymbol{Z}^n\} = \min\{y^T b | y^T A = c^T, y \in \boldsymbol{Z}^m_+\},$$

falls die zugehörigen LP endlich lösbar sind.

Beweis. $\{x | Ax \leq b\}$ ganz, $\{y | y^T A \leq c^T, y^T(-A) \leq -c^T, -Ey \leq 0\}$ zusammengesetzte vollständig unimodulare Restriktionsmatrix $\rightsquigarrow$ ganz $\rightsquigarrow$ Seiten der optimalen (rationalen) Lösung enthalten ganzzahlige Punkte. □

Satz 5.3 *A ganz mit vollem Zeilenrang. Dann ist A unimodular genau dann, wenn*

$$P(b) = \{x | Ax = b, x \geq 0\} \quad \textit{ganz für alle } b \in \boldsymbol{Z}^m.$$

Beweis. "$\Rightarrow$" Sei A unimodular, b ganz, x Ecke von $P(b)$. $x_B = A_B^{-1}b, x_N = 0 \rightarrow x$ ganz.

"$\Leftarrow$" Sei nun $P(b)$ ganz für alle $b \in \boldsymbol{Z}^m$. Sei A_B reguläre $m \times m$ Teilmatrix. Zu zeigen genügt: $A_B^{-1}t$ ganz für alle ganzen t (z.B. für die Einheitsvektoren e_i). Wähle $y = y(t)$ ganz mit $z := y + A_B^{-1}t \geq 0 \rightsquigarrow b := A_B z = A_B y + t$ ganz. Setze $x_B := A_B^{-1}b = z, x_N = 0 \overset{P(b) \text{ ganz}}{\rightsquigarrow} x$ ganz $\rightsquigarrow z$ ganz $\rightsquigarrow A_B^{-1}t = z - y$ ganz. □

Satz 5.4 *A ganz. Dann ist A vollständig unimodular*

$$\begin{aligned} &\Leftrightarrow \quad P(B) = \{x|Ax \leq b, x \geq 0\} \quad \textit{ganz für alle } b \in \boldsymbol{Z}^n \\ &\Leftrightarrow \quad \{x|a \leq Ax \leq b, c \leq x \leq d\} \quad \textit{ganz für alle ganzen Vektoren } a, b, c, d \end{aligned}$$

Beweis.

1. Äquivalenz: $P(b)$ ganz für alle $b \in \boldsymbol{Z}^m$

$$\begin{aligned} &\Leftrightarrow \quad \{x|Ax + Ey = b, x \geq 0, y \geq 0\} \quad \text{ganz für alle } b \in \boldsymbol{Z}^n \\ &\Leftrightarrow \quad [A|E] \quad \text{unimodular} \\ &\Leftrightarrow \quad A \text{ vollständig unimodular} \end{aligned}$$

2. Äquivalenz: $[A|-A|E|-E]^T$ vollständig unimodular $\leftrightarrow$ A vollständig unimodular.□

Bemerkung:

1. Der Satz 5.4 liefert ganzzahlige Dualitätssätze analog Korollar 5.2.
2. Gelten diese für alle Parameter, so ist A vollständig unimodular. z.B.

   $$\max\{c^T x|Ax \leq b, x \in \boldsymbol{Z}_+^n\} = \min\{y^T b|y^T A \geq c^T, y \in \boldsymbol{Z}_+^m\}$$

   für alle b, c ganz, für die die zugehörigen LP endlich lösbar sind.

Satz 5.5 *A $0, \pm 1$ Matrix. Dann sind äquivalent:*

1. *A vollständig unimodular.*
2. *(Ghouila-Houri 62): Jede Spaltenmenge J besitzt eine Zerlegung $S \cup T$ mit $\sum_{j \in S} A_j - \sum_{j \in T} A_j$ $0, \pm 1$ Vektor.*
3. *(Camion 63/65): Jede quadratische nichtsinguläre Teilmatrix B besitzt eine Zeile mit ungerader Anzahl von Nichtnullelementen.*
4. *(Camion 63/65): Die Summe aller Elemente jeder quadratischen Teilmatrix mit geraden Zeilen- und Spaltensummen ist durch 4 teilbar.*
5. *(Gomory) Keine quadratische Teilmatrix hat Determinante ± 2.*

Beweis. $1 \rightarrow 2 \begin{array}{l} \nearrow 3 \\ \searrow 4 \end{array} \left.\vphantom{\begin{array}{l} 3 \\ 4 \end{array}}\right\} \rightarrow 5 \rightarrow 1.$

$1 \to 2$: Sei $d_j := \begin{cases} 0 & j \notin J \\ 1 & j \in J \end{cases}$. Aus Satz 5.4 folgt

$$P := \left\{ x | 0 \leq x \leq d, \underbrace{\left\lfloor \frac{1}{2} A d \right\rfloor}_{a} \leq Ax \leq \underbrace{\left\lceil \frac{1}{2} A d \right\rceil}_{b} \right\} \quad \text{ganz.}$$

$\frac{1}{2}d \in P$, $\frac{1}{2}Ad - a = b - \frac{1}{2}Ad$ ist $0, \frac{1}{2}$ Vektor (*).

Sei x Ecke von $P \to x$ $0,1$–Vektor, $(Ax)_i \in \{a_i, b_i\}$ für alle i. Sei $y := d - 2x \to Ay = 2[\frac{1}{2}Ad - Ax]$ ist wegen (*) $0, \pm 1$ Vektor $\to y_j = 0$ $(j \notin J)$, $y_j \in \{\pm 1\}$ $(j \in J)$. Sei $S := \{j | y_j = 1\}$, $T := \{j | y_j = -1\} \to Ay = \sum_{j \in S} A_j - \sum_{j \in T} A_j$ ist $0, \pm 1$ Vektor, das heißt es gilt 2).

Bemerkung: Sei Q quadratische $0, \pm 1$ Matrix. Dann sind äquivalent

(a) $Q1$ gerade.

(b) Qx gerade für alle $0, \pm 1$–Vektoren.

(c) $\det Q$ gerade.

Beweis. a) bedeutet: Anzahl von ± 1 Elementen ist gerade in jeder Zeile, also b). a) bedeutet in der linearen Algebra: Spalten linear abhängig über GF(2) also c).

$2 \to 3$: B quadratische Teilmatrix mit **gerader** Anzahl von ± 1 Elementen in jeder Zeile. also Bx gerade für alle $0, \pm 1$ Vektoren. Aus 2) folgt By ist $0, \pm 1$ Vektor für einen ± 1 Vektor y also $By = 0 \overset{y \neq 0}{\to} B$ ist singulär.

$2 \to 4$: B quadratische Teilmatrix mit geraden Zeilen/Spaltensummen. Analog (Schritt $2 \to 3$) folgt aus 2) $\sum_{j \in S} B_j = \sum_{j \in T} B_j$ für Zerlegung $S \cdot \cup T$ der Spalten B. Sei $\sum Q =:$ Summe aller Elemente einer Matrix Q. Da die Spaltensummen gerade sind, sind auch $\sum B_S$ und $\sum B_T$ gerade. $\sum B_S = \sum B_T \to 4$ teilt $\sum B$.

$3 \vee 4 \to 5$: Induktionsannahme: 1–5 äquivalent für **echte** Teilmatrizen von A. Zu zeigen für A quadratisch: $3 \vee 4 \Rightarrow \det A \notin \{\pm 2\}$. Annahme: $\det A \in \{\pm 2\}$. Aus der obigen Bemerkung folgt $A1$ gerade $\to$ 3) falsch, also Annahme 4) gültig. Sei

$$A =: \begin{bmatrix} a_1 \\ B \end{bmatrix} =: \begin{bmatrix} A_1 & \begin{matrix} d \\ C \end{matrix} \end{bmatrix}$$

Wegen 4) gilt 2) für B, das heißt $Bx = 0$ für einen ± 1 Vektor x und 3) gilt für C, das heißt $\det C$ ungerade.

$$\to A \begin{bmatrix} & 0 & \ldots 0 & \\ x & 1 & & \\ & & \ddots & \\ & & & 1 \end{bmatrix} = \begin{bmatrix} \alpha & d \\ 0 & C \end{bmatrix}$$

$$\rightarrow \underbrace{\det A \cdot \pm 1}_{\pm 2} = \underbrace{\alpha}_{\text{gerade}} \cdot \underbrace{\det C}_{\text{ungerade}} \rightarrow |\alpha| = |\det A|$$

Wegen der obigen Bemerkung gilt $1^T A$ gerade. Zusammen mit dem Resultat, daß $1 - x$ gerade ist, denn x ist ± 1 Vektor, folgt $1^T A(1 - x) \equiv 0 \bmod 4$. Zusammen mit $1^T Ax = \alpha \equiv 2 \bmod 4$ folgt schließlich $1^T A1 \equiv 2 \bmod 4$. Dies ist ein Widerspruch zu 4).

$5 \rightarrow 1$: zu zeigen; jede quadratische $0, \pm 1$ Matrix B mit $|\det B| > 2$ besitzt quadratische Teilmatrix mit $\det \in \{\pm 2\}$. Sei B $n \times n$. $C := [B\ E]$. Durch Addition und Subtraktion von Zeilen und Multiplikation von Spalten mit -1 erhalten wir C' mit folgenden Eigenschaften:

(a) $0, \pm 1$ Matrix.

(b) E Teilmatrix (permutiert).

(c) maximale Anzahl Einheitsspalten in Spalte $1 \ldots n$.

O. B. d. A. seien die Spalten $1 \ldots n$ und $n+1 \ldots 2n$ so angeordnet daß C' die folgende Form habe (Spaltenpermutation)

$$C' = \begin{bmatrix} E_k & & X_k & \\ & D_{n-k} & & E_{n-k} \end{bmatrix}$$

Die Determinate der ersten n Spalten hat den betragsmäßig gleichen Wert wie $\det B$. Wegen der Annahme $\det B \in \{\pm 2\}$ muß $k < n$ gelten. Spalte j **nicht** in Einheitsspalte transformierbar (das heißt die Elimination mindestens eines weiteren ± 1 Elements in der Spalte j ist nicht möglich) ohne i) zu verletzen. Das heißt es gibt folgende Konstellation in C' (in den Spalten $k + 1 \ldots, k + n$:

$$\begin{array}{ccc} & j & j' \\ i & 1 & +1 \\ i' & 1 & -1 \end{array} \quad \text{bzw.} \quad \begin{array}{ccc} & j & j' \\ i & 1 & -1 \\ i' & -1 & -1 \end{array}$$

Die Teilmatrizen haben jeweils $|\det| = 2$. $C'_j, C_{j'}$ zu $n \times n$ Matrix H' mit Einheitsspalten $\neq i, i'$ erweitern $\rightarrow |\det H'| = 2$. Sei H die entsprechende Matrix in $C \rightarrow |\det H| = 2$, da H_μ für $\mu > n$ Einheitsspalte $\rightarrow B$ enthält Teilmatrix mit $|\det| = 2$. Widerspruch zur Annahme.

Korollar 5.6 *1. (Heller, Tompkins, Gale 56): A sei* $0, \pm 1$ *Matrix mit genau zwei Elementen* $\neq 0$ *pro Spalte. Dann ist A vollständig unimodular genau dann, wenn man die Zeilen in zwei Klassen aufteilen kann, so daß gilt:*

* *zwei nicht verschwindende Elemente einer Spalte gehören zu verschiedenen Klassen genau dann, wenn sie gleiches Vorzeichen haben.*

2. *(Poincare, 1900): A sei* $0, \pm 1$ *Matrix mit genau einer* $+1$ *und einer* -1 *pro Spalte. Dann ist A vollständig unimodular.*

Beweis.

1. Satz 5.5 (2) für A^T:

   A vollständig unimodular $\Leftrightarrow$ es existieren Klassen S, T mit $\sum\limits_{j \in S} a_j - \sum\limits_{j \in T} a_j$ $0, \pm 1$ Vektor.

$$\begin{array}{ccc|ccc} & & 1 & & & \\ 1 & -1 & -1 & 1 & & 1 \\ \hline 1 & -1 & & -1 & -1 & \\ & & & & -1 & \end{array} \Leftrightarrow (*) \text{ für } S \text{ und } T.$$

2. folgt aus 1. mit *einer* Klasse $\hat{=}$ Menge aller Zeilen. $\square$

Bemerkung:

1. Bedingung von HTG ist effizient nachprüfbar.
2. Allgemein: vollständig unimodular ist effizient überprüfbar, aber das Verfahren ist sehr komplex. (s. Buch von Schrijver; Arbeiten von Seymour (1980) und Truemper (1987)).

Beispiel Gerichtete Graphen:

$G = (V, E)$ gerichteter Graph mit Inzidenzmatrix $A(G)$ $|V| \times |E|$

$$a_{k\,ij} := \begin{cases} -1 & k = i \\ +1 & k = j \\ \ \ 0 & \text{sonst} \end{cases}$$

$\overset{5.6\,2.}{\rightsquigarrow}$ $A(G)$ vollständig unimodular $\rightsquigarrow$ ganzzahlige Dualitätssatz für Flußprobleme:

$$\min\{a^T x | Ax = 0, l \leq x \leq c, x \in \boldsymbol{Z}^n\} = \max\{\lambda^T l - \gamma^T c | uA + \lambda^T - \gamma^T \leq a^T, \lambda \geq 0, \gamma \geq 0, u, \lambda, \gamma \text{ganz}\}$$

Beispiel Bipartite Graphen:

$G = (V, E)$ ungerichteter, bipartiter Graph mit Inzidenzmatrix $M(G)$ $|V| \times |E|$

$$m_{k\,ij} := \begin{cases} 1 & k = i \vee k = j \\ 0 & \text{sonst} \end{cases}$$

$\overset{\text{HTG}}{\rightsquigarrow}$ $M(G)$ vollständig unimodular.

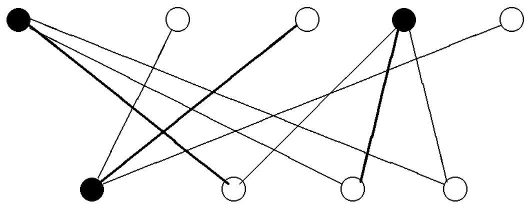


Abbildung II.34: maximales Matching und minimale Knotenüberdeckung

Ganzzahlige Dualitätssätze für bipartite Graphen

$$\underbrace{\max\{\sum x(e) | Mx \leq 1, x \in \boldsymbol{Z}_+^{|E|}\}}_{\text{maximales Matching}} = \underbrace{\min\{\sum y(i) | y^T M \geq 1, y \in \boldsymbol{Z}_+^{|V|}\}}_{\text{minimale Knotenüberdeckung}}$$

$$\underbrace{\min\{\sum x(e) | Mx \geq 1, x \in \boldsymbol{Z}_+^{|E|}\}}_{\text{minimale Kantenüberdeckung}} = \underbrace{\max\{\sum y(i) | y^T M \leq 1, y \in \boldsymbol{Z}_+^{|V|}\}}_{\text{maximale stabile Menge}}$$

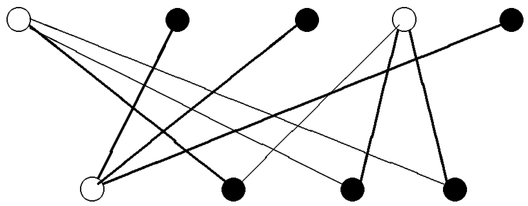
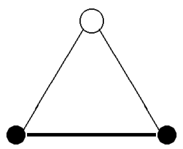


Abbildung II.35: minimale Kantenüberdeckung und maximale stabile Menge

Beispiel Ungerichtete Graphen:



nicht bipartit

Abbildung II.36: Gegenbeispiel für ganzzahligen Dualitätssatz

maximales Matching $= 1 \neq 2 =$ minimale Knotenüberdeckung. (vgl. Abbildung II.36)

$$\det M = \det \begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix} = 2$$

6 Vollständig Dual Ganzzahlige Systeme

1. $P = \{x|Ax \leq b\}$, A, b rational, fest.
2. $z(c) := \max\{c^T x | x \in P\}$ für $c \in \boldsymbol{Q}^n$.
3. $C := \{c | z(c) \text{ endlich } \}$
4. $S(c) := \{x \in P | c^T x = z(c)\}$ optimale Seite zu $c \in C$.

Seiten

Lemma 6.1

1. *H Stützhyperebene $\Rightarrow H \cap P = S(c)$ für ein $c \in C$.*
2. *Jede Seite ist optimale Seite für ein $c \in C$*
3. *Enthält jede optimale Seite einen ganzzahligen Punkt, so ist P ganz.*
4. *Minimale Seiten sind Lösungsmengen eines Teilsystems $A'x = b'$ von $Ax \leq b$.*

Beweis.

1. $H \cap P \neq \emptyset$, $P \subseteq H_{\leq}$, $H = \{x \in P | c^T x = \delta\}$ $(c, \delta$ rational). $z(c) = \delta$, $c \in C \leadsto$ $H \cap P = \{x \in P | c^T x = \delta\}$.
2. $\emptyset \neq S = \{x \in P | A'x = b'\}$ für ein Teilsystem $A'x = b'$.
   Ansatz $c := 1^T A'$. Sei $x \in P$, $y \in S$: $c^T x = 1^T A' x \leq 1^T A' y = c^T y \leadsto S = S(c)$.
3. trivial aus 2.
4. siehe Kapitel I. □

Ganzzahlige Gleichungen

Lemma 6.2 *Hermite'sche Normalform: Sei A von maximalen Rang, $n \geq m$. Dann gibt es eine unimodulare Matrix U mit $AU = [B0]$, wobei B reguläre untere Dreiecksmatrix, $B \geq 0$, $b_{ij} < b_{ii}$ für $j < i$.*

Beweis. siehe Lineare Algebra, Buch von Schrijver. □

Lemma 6.3 $\exists\, x$ *ganz* $Ax{=}b \Leftrightarrow \quad \forall\, y$ *rational* $(y^T A$ *ganz* $\Rightarrow y^T b$ *ganz*$)$

Beweis. "$\Rightarrow$" $\underbrace{y^TA}_{\text{ganz}}\underbrace{x}_{\text{ganz}} = \underbrace{y^Tb}_{\text{ganz}}$

"$\Leftarrow$" Entweder rational lösbar oder genau $u^TA = 0$, $u^Tb = 1$ für ein rationales u lösbar. Zweite Alternative geht nicht, da für $y = \frac{1}{2}u$ obige Bedingung verletzt, daher OBdA Zeilen linear unabhängig (in rechter Seite übrige $y_i = 0$ wählen) daher OBdA $A = [B0]$ in Hermitescher Normalform (denn $Ax = b \Leftrightarrow AU(U^{-1}x) = b$).

Ansatz $\begin{pmatrix} x \\ z \end{pmatrix} = \begin{bmatrix} B^{-1}b \\ 0 \end{bmatrix}$, z ganz.

Ansatz für y-Vektoren: $Y^T := B^{-1} : Y^TA = B^{-1}[B0] = [E0]$ ganz $\rightsquigarrow x = Y^Tb$ ganz. □

Satz 6.4 (Edmonds, Giles 77) *P ganz genau dann, wenn jede Stützhyperebene einen ganzzahligen Punkt enthält.*

Beweis. "$\Rightarrow$" wegen Lemma 6.1 (1) und (3).

"$\Leftarrow$" Annahme: minimale Seite $S = \{x|A'x = b'\} \cap \boldsymbol{Z}^n = \emptyset \rightsquigarrow c^T = y^TA'$ ganz, $\delta = y^Tb' \notin \boldsymbol{Z}$ für ein rationales y (OBdA $y > 0$, sonst $y := y + u$ mit $u > 0$ ganz). Sei $H = \{x|c^Tx = \delta\}$, $x \in P$, $x' \in S$. $c^Tx = y^TA'x \leq y^Tb' = y^TA'x' = c^Tx'$. $H = S(c)$ Stützhyperebene. z ganz $\rightsquigarrow c^Tz$ ganz $\rightsquigarrow c^Tz \neq \delta \Rightarrow H \cap \boldsymbol{Z}^n = \emptyset$. □

Korollar 6.5 *P ganz $\Leftrightarrow z(c) \in \boldsymbol{Z}$ für alle $c \in C \cap \boldsymbol{Z}^n$.*

Beweis. "$\Rightarrow$" P ganz, $c \in C \cap \boldsymbol{Z}^n \rightsquigarrow S(c) \cap \boldsymbol{Z}^n \neq \emptyset \rightsquigarrow z(c) \in \boldsymbol{Z}$.

"$\Leftarrow$" zu zeigen: Bedingung in Satz 6.4. $H = \{x|c^Tx = \delta\}$ Stützhyperebene für $\delta = z(c)$. OBdA $c_1, \ldots, c_n \in \boldsymbol{Z}$ teilerfremd. Nach Voraussetzung $z(c) \in \boldsymbol{Z}$.

Annahme: $H \cap \boldsymbol{Z}^n = \emptyset \overset{\text{Lemma 6.2}}{\rightsquigarrow} \exists\, y_0 \in \boldsymbol{Q} : \underbrace{y_0c^T \in \boldsymbol{Z}^n}_{y_0 \in \boldsymbol{Z}}, \underbrace{y_0\delta \notin \boldsymbol{Z}}_{y_0\delta \in \boldsymbol{Z}} \Rightarrow H \cap \boldsymbol{Z}^n \neq \emptyset$. □

Vollständig dual ganzzahlige Systeme

$Ax \leq b$ rational heißt VDG-System, falls für alle $c \in C \cap \boldsymbol{Z}^n$ eine ganze OL y_* zu $\min\{y^Tb|y^TA = c^T, y \geq 0\}$ existiert.

Satz 6.6 *$Ax \leq b$ VDG-System, b ganz $\Rightarrow P = \{x|Ax \leq b\}$ ganz.*

Beweis. Zu zeigen: $z(c)$ ganz für alle $c \in C \cap \boldsymbol{Z}^n$. Zu c gibt es eine ganz duale OL y_*, also $z(c) = y_*^Tb$ ganz. □

Korollar 6.7 *$Ax \leq b$ VDG-System, b ganz. Dann gilt für $c \in C \cap \boldsymbol{Z}^n$*

$$\max\{c^Tx|Ax \leq b, x \in \boldsymbol{Z}^n\} = \min\{y^Tb|y^TA = c, y \in \boldsymbol{Z}^n_+\}$$

Beispiel:

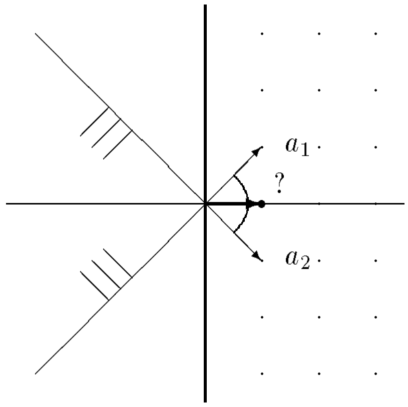


Abbildung II.37: Zu Beispiel 1

1. $\begin{array}{l} x_1 + x_2 \leq 0 \\ x_1 - x_2 \leq 0 \end{array}$ ist kein VDG-System, denn $y_1 a_1 + y_2 a_2 \neq (1,0)^T \quad \forall\, y \in \boldsymbol{Z}_+^2$.
   Mit der zusätzlichen Restriktion $x_1 \leq 0$ ist es ein VDG-System.

2. A vollständig unimodular $\leadsto$ $Ax \leq b$ VDG-System.

3. $\underbrace{\partial x(S) \leq h(S) \text{ für alle } S \subseteq V,\ l \leq x \leq u}_{\text{submodulares Flußpolyeder}}$ ist VDG-System, falls $h : S \mapsto h(S)$ submodulare Funktion ist:

$$h(S) + h(T) \geq h(S \cap T) + h(S \cup T) \quad \forall\, S, T \subseteq V$$

   Falls h ganz, so sind submodulare Flußpolyeder ganz.

4. $z = \min\{\sum x(e) | Mx \geq 1, x \in \boldsymbol{Z}_+^6\}$

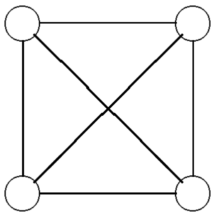
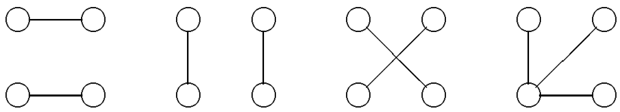


Abbildung II.38: Zu Beispiel 4

$\leadsto$ Polyeder ist ganz.

Duales Problem: $\max\{\sum y(j) | y^T M \leq 1, y \geq 0\} = 2$ aber einzige OL ist $y_* \equiv \frac{1}{2} \leadsto$ $Mx \geq 1$ kein VDG-System.



und symmetrisch

insgesamt 7 Ecken (0,1-Ecken)

Abbildung II.39: Eckenliste des primalen Polyeders

7 Schnittebenenverfahren

(7.1)
$$\min\{c^T x \mid \underbrace{Ax = b, x \in \boldsymbol{Z}_+}_{P}\},\; A, b, c \text{ ganz}$$

Gomory 58

Schnitte zu LP adjungieren.

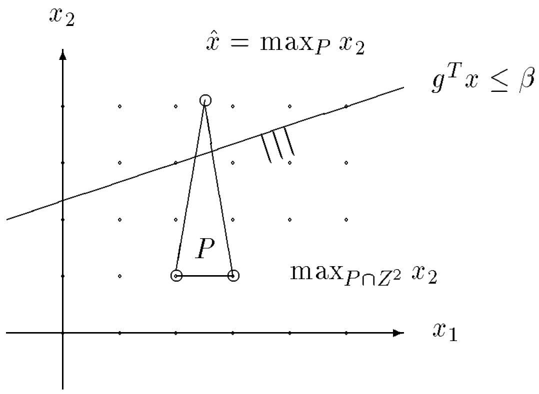


Abbildung II.40: Gomory-Schnitte zum LP adjungieren

Schnitt aus Tableau des SV

$$\begin{aligned}
x_{B(i)} - \sum t_{ij} x_{N(j)} &= t_{i0} \notin \boldsymbol{Z} \quad \text{für ein } i \in \{0, \ldots, m\} \\
x_{B(i)} - \sum \lceil t_{ij} \rceil x_{N(j)} &\leq \lfloor t_{i0} \rfloor \quad \text{da } x \geq 0 \text{ und ganz} \\
\sum \underbrace{(\lceil t_{ij} \rceil - t_{ij})}_{0 \leq f_{ij}} x_{N(j)} &\geq \underbrace{(t_{i0} - \lfloor t_{i0} \rfloor)}_{1 > f_{i0} > 0}
\end{aligned}$$

(7.2)
$$s + \sum (-f_{ij}) x_{N(j)} = -f_{i0}, \quad s \in \boldsymbol{Z}_+$$

s ganz:

$$s = -f_{i0} - \sum (-f_{ij}) x_{N(j)}$$

$$= \quad \lfloor t_{i0} \rfloor - \sum(\lceil t_{ij} \rceil) x_{N(j)} - x_{B(i)}$$

(7.2) ist Schnitt: $\hat{x}$ einsetzen, $\hat{x}_N = 0$

$$s = -f_{i0} < 0 \leadsto \hat{x} \text{ unzulässig}$$

Algorithmus 7.1 (Das Gomory-Verfahren)

1. Bestimme OL $\hat{x}$ zu LP.
2. Ist $\hat{x}$ ganz, STOP.
3. Wähle i minimal mit $f_{i0} > 0$; adjungiere (9.2) am Tableauende.
4. Bestimme OL $\hat{x}$ mit dem dualem Simplexverfahren (*) und gehe nach 2).

(*)-Zusatzregeln

1. Lexikographische Spaltenauswahl s zu Pivotzeile r:

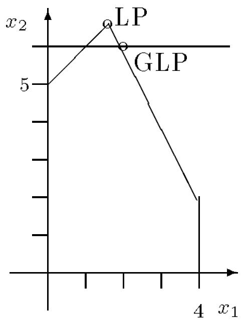
$$s : \text{lexmax}\{ \begin{pmatrix} \vdots \\ t_{ij}/t_{rj} \\ \vdots \end{pmatrix} \mid t_{rj} < 0\}$$

2. Eliminiere einen Schnitt, sobald seine Schlupfvariable wieder in die Basis eintritt.

Beispiel

$$\begin{array}{lrcrcr} \text{max} & x_1 & + & 2x_2 & & \\ \text{unter} & 2x_1 & + & x_2 & \leq & 10 \\ & -x_1 & + & x_2 & \leq & 5 \end{array}$$

$0 \leq x_1 \leq 4, 0 \leq x_2, x_1 \in \boldsymbol{Z}, x_2 \in \boldsymbol{Z}$



Optimales Tableau zu (LP)

		x_3	x_4
	15	-1	-1
x_1	$5/3$	$-1/3$	$1/3$
x_2	$20/3$	$-1/3$	$-2/3$
s_1	$-2/3$	$1/3$	$2/3$

$\leftarrow f$ $\boxed{2/3 \mid -1/3 \quad -2/3}$ $\Leftrightarrow x_2 \leq 6$

Dualer Simplexschritt

		x_3	s_1
	14	$-1/2$	$-3/2$
x_1	2	$-1/2$	$1/2$
x_2	6	0	-1
x_4	1	$-1/2$	$3/2$

optimal, zulässig
$x_1 = 2 \leq 4$
$x_2 = 6$
$c^T x = 14$

Satz 7.1 (Gomory) *Sei LP beschränkt. Dann liefert das Gomory-Verfahren nach endlich vielen Schritten eine OL des Problems (7.1).*

Beweis.

1. Tableau zu $\hat{x}$ hat $t_{0j} \geq 0$ (min-Problem). OBdA Allle Spalten lexpos (sonst $\sum x_{N(j)} \leq p$ zwischen 0-ter und 1-ter Zeile einfügen.)

2. Im Schritt 3 des Verfahrens bleiben Spalten lexpos, da neue Restriktion unten angefügt wird.

3. Bei dualen Pivotschritten bleiben Spalten lexpos, da mit lexikographischer Zusatzregel iteriert wird. Bei Elimination eines Schnittes bleiben Spalten lexpos.

4. Spalte t_{i0} fällt streng lexikographisch. Bei Elimination eines Schnittes ebenfalls.

5. Annahme: nicht endlich
   $t_{i0}^{(k)}$ streng fallend $\rightsquigarrow \quad \exists\, d : \; t_{00}^{(k)} \searrow d$. Etwa nach q Iterationen gilt

   $$t_{00}^{(q)} = \lfloor d \rfloor + f \;\text{ mit } 0 \leq f < 1$$

   Annahme: $f > 0$

   $$\boxed{-f \mid \ldots - f_{0j} \ldots} \text{ ist Schnitt}$$

   Pivot mit $-f_{0s} < 0$ (oder Abbruch mit $P = \emptyset$)

   $$d \leq t_{00}^{(q+1)} = t_{00}^{(q)} - \frac{-f t_{0s}}{-f_{0s}} = t_{00}^{(q)} - f \underbrace{\frac{t_{0s}}{f_{0s}}}_{t_{0s} \geq f_{0s} > 0} \leq t_{00}^{(q)} - f = \lfloor d \rfloor \leq d$$

   $\rightsquigarrow t_{00}^{(q+1)} = d$ ganz nach endlich vielen Schritten. Anschließend ist $t_{00}^{(k)}$ fix für $k \geq q+1$.

6. Analog für weitere Zeilen $t_{10}^{(k)}, t_{20}^{(k)}, \ldots$ sukzessiv jeweils nach endlich vielen Schritten.

7. Nie mehr als n Variablen in der Basis. $\square$

Bemerkung: Generell anwendbar, aber langsam: $O((p+1)^{n+1} d \sum_{j=1}^{n}(|c_j|+1))$ siehe Nemhauser, Wolsey S.373

Rundreisepolytop Q

$G = K_n$ vollständig, ungerichtet, n Knoten.

$$x^R(ij) := \begin{cases} 0 & ij \notin R \\ 1 & ij \in R \end{cases}$$

$Q = \operatorname{conv}\{x^R | R \text{ Rundreise}\} \subseteq [0,1]^E$.

$$z_{\text{TS}} = \min\{c^T x | x \in Q\} \tag{7.3}$$

2-Matching

$x^R \to x^R(\delta(i)) = 2 \quad \forall\, i \in V$. Aber Teilrundreisen so erlaubt (s. Abbildung II.41).

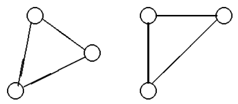


Abbildung II.41: Teilrundreise bei 2-Matchings

Es gilt:

1. $\operatorname{aff}(Q) = \{x | x(\delta(i)) = 2 \quad \forall\, i \in V\}$
2. $\dim Q = |E| - |V|$ siehe Nemhauser, Wolsey prop 3.4.

Facetten

Triviale Facetten: $0 \leq x(e) \leq 1 \quad \forall\, e \in E$ (Facetten für $n \geq 5$).

Elimination von Teilrundreisen

$E(W) := \{ij \in E | i, j \in W\}$ für $W \subseteq V$.

$$x(E(W)) \leq |W| - 1 \text{ für } \emptyset \neq W \subset V \tag{7.4}$$

ist Facette für $2 \leq |W| \leq n-2$, $n \geq 6$.

(7.5) 2 solche Facetten zu W, W' sind genau dann äquivalent, falls $W' = W \setminus V$

Anzahl : $2^{n-1} - n - 1$.

LP-Relaxation zum Rundreiseproblem

$$\min\{c^T x \mid \underbrace{x(\delta(i)) = 2,\ i \in V,\ 0 \le x \le 1\}}_{P} \le z_{\mathrm{TS}} \tag{7.6}$$

Lemma 7.2 *Sei $x \in P$. Dann gilt:*

$$x(E(W)) = |W| - 1 + \epsilon \Leftrightarrow x(\delta(W)) = 2 - 2\epsilon$$

Beweis. $2|W| = \sum_{i\in W} x(\delta(i)) = 2x(E(W)) + x(\delta(W)) \leadsto 2 - x(\delta(W)) = 2(x(E(W)) - (|W| - 1))$ □

Folgerung: Sei x^* OL zu (7.6). x^* erfüllt (7.4) $\Leftrightarrow$

$$x^*(\delta(W)) \ge 2 \ \ (W \subseteq V, 3 \le |W| \le n-3) \tag{7.7}$$

($|W| = 2$ trivial wegen $x(e) \le 1$. $|W| = n-2$ symmetrisch)

Facettengenerierung

$$\min\{x^*(\delta(W)) | W \subseteq V, 3 \le |W| \le n-3, 1 \in W\} \tag{7.8}$$

($1 \in W$ OBdA, da $x^*(\delta(W)) = x^*(\delta(\bar{W}))$ und $1 \in W$ oder $1 \in \bar{W}$)

Lösung des Facettengenerierungsproblems

Sei W^* optimal $\leadsto$ existiert j minimal mit $1, \ldots, j-1 \in W^*, j \notin W^*, j \le n-2$. Löse für $j = 2, \ldots, n-2$ die Minimum-1-j-Schnitt-Probleme

$$\lambda_j := \min\{x^*(\delta(W')) | \{1, \ldots, j-1\} \subseteq W, j \in \bar{W}, 3 \le |W| \le n-3\}$$

x^* definiert Kapazitäten, setze $x^*(1k) = \infty$ für $2 \le k \le j-1$)

Folgerung:
$\min \lambda_j < 2 \leadsto W^*$ definiert eine neue Facette.
$\min \lambda_j \ge 2 \leadsto$ keine Restriktion vom Typ (7.4) verletzt.

Kamm-Ungleichungen

$|H \cap T_i| \ge 1 \quad \forall\, i,$
$|T_i \setminus H| \ge 1 \quad \forall\, i,$
$|T_i \cap T_j| = \emptyset \quad \forall\, i \ne j.$
($k \ge 3$, k ungerade)

Satz 7.3 (Grötschel/Padberg 77) *Dann ist*

$$x(E(H)) + \sum_i x(E(T_i)) \le |H| + \sum_i (|T_i| - 1) - \frac{k+1}{2} \tag{7.9}$$

eine Facette des Rundreisepolytopen für $n \ge 6$.

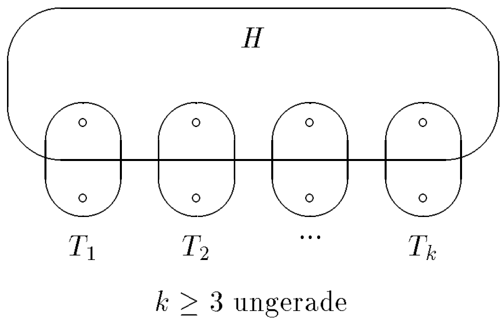


Abbildung II.42: Kammungleichung mit Griff H und Zähnen T_i

Lemma 7.4 *(7.9) wird von jeder Rundreise erfüllt.*

Beweis. Sei x Rundreise. Dann gilt

$$\begin{aligned}
& 2[x(E(H)) + \sum_i x(E(T_i))] \\
\leq\ & \sum_{v \in H} x(\delta(v)) + \sum_i [x(E(T_i)) + x(E(T_i \setminus H)) + x(E(T_i \cap H))] \\
\leq\ & 2|H| + \sum_i [(|T_i| - 1) + (|T_i \setminus H| - 1) + (|T_i \cap H| - 1)] \\
=\ & 2[|H| + \sum_i (|T_i| - 1)] - k
\end{aligned}$$

Rest klar, da linke Seite der behaupteten Ungleichung ganz □

Beispiel:

Untere Schranke für die Anzahl der Facetten von Q

Anzahl der Städte	10	20	30	40	50	120
Anzahl der nichtäquivalenten Kammungleichungen	8841970	$1,5 \cdot 10^{18}$	$1,5 \cdot 10^{31}$	$1,5 \cdot 10^{45}$	10^{60}	$2 \cdot 10^{179}$

Grötschel, Padberg (1979)

Schnittebenenverfahren

Dantzig, Fulkerson, Johnson (1954), Grötschel (1980):

$$Q \subseteq P \begin{cases} 0 \leq x \leq 1 \\ x(\delta(i)) = 2 \quad i \in \{1, 2, \ldots, n\} \end{cases}$$

$$(LP)_P \qquad \min\{\sum_{i<j} c_{i,j} \cdot x_{i,j} \mid x \in P\}$$

Algorithmus

1. Schritt: Löse $(LP)_P$ mittels Simplexverfahren.

2. Schritt: Ist die optimale Ecke $\hat{x}$ eine Rundreise, dann STOP.

3. Schritt: Wähle eine Anzahl geeigneter Facetten $[Bx \leq b]$ mit $B\hat{x} > b$.

4. Schritt: $P := P \cap \{x \in \boldsymbol{R}^{\frac{1}{2}n(n-1)} \mid Bx \leq b\}$. Gehe nach 1.

120-Städte-Problem

Grötschel (1980):
13 Iterationen
$13x(\frac{1}{2} - 3h)$
96 Facetten

Numerische Untersuchungen

Padberg, Hong (1980)
u.a. 318 St., $\frac{d(\pi)}{z^*} \leq 1,0025$

Linienoptimierung

Gegeben sei ein Eisenbahnnetz mit periodischen Fahrplänen, also Fahrplänen, bei denen eine Linie innerhalb einer Zeiteinheit wie z.B. einer Stunde in einem festen Takt (z.B. alle 12 Minuten) bedient wird. Solche Fahrpläne werden bei fast allen städtischen öffentlichen Verkehrssystemen, wie z.B. Straßenbahnen oder Bussen, aber auch zunehmend bei Eisenbahngesellschaften, wie z.B. der Nederlandse Spoorwegen oder auch der Deutschen Bahn insbesondere bei ICE/IC- und IR-Netzen benutzt.
Ein solches Eisenbahnnetz kann unter verschiedenen Gesichtspunkten optimiert werden. Zum einen kann versucht werden, die Betriebskosten bei vorgegebener Service-Qualität zu minimieren, zum anderen kann man auch die Betriebskosten vorgeben, und versuchen, die Service-Qualität (Wartezeiten, Reisegeschwindigkeit etc.) zu optimieren.
Vereinfacht kann ein Eisenbahnnetz durch einen ungerichteten Graphen $G = (V, E)$ modelliert werden, dessen Knotenmenge V und Kantenmenge E die Stationen respektive die Schienenstränge zwischen den Stationen des Netzes darstellen. Die möglichen Linien des Netzes sind dann die (einfachen) Wege in G. Da die Bahnhöfe, in denen Linien beginnen oder enden können, besonders ausgestattet sein müssen (z.B. durch zusätzliche Rangiermöglichkeiten), bilden diese eine Teilmenge $\mathcal{CY} \subseteq V$.
Zu einem Paar $t := (a, b)$ Stationen $(a, b \in V, a \neq b)$ bezeichne $tr(t)$ die Anzahl der Reisenden von a nach b. $\mathcal{T} := \{(a, b)|a, b \in V, a \neq b, tr((a, b)) \neq 0\}$ ist dann die Menge der Paare von Bahnhöfen, für die überhaupt ein Verkehrsbedarf besteht.
Die Gesamtheit der möglichen Linien innerhalb des untersuchten Netzes werden mit $\mathcal{L}$ bezeichnet, eine Linie zwischen zwei Stationen mit $l \in \mathcal{L}$ ("line"). Eine solche Linie l wird innerhalb einer festen Zeiteinheit, etwa einer Stunde, in einem festen Takt, wie z.B. alle

15 Minuten bedient. Diese Freqenz wird mit $f(l) \in \boldsymbol{Z}_+$ bezeichnet.
Um das Modell zu vereinfachen, wird angenommen, daß Reisende zwischen zwei Stationen immer einen kürzesten Weg nutzen (bezüglich einer Kantenbewertung, etwa Fahrzeit, -preis etc). Diese Voraussetzung ist zumindestens für großflächige Verkehrsnetze realistisch.
Da die Zugkapazitäten fest vorgegeben sind, läßt sich die Linienbedarfsfrequenz $lfr(e)$ ("line-frequency-requirement") für jede Kante $e \in E$ leicht berechnen.

In der Realität ist die Größe $lfr(e)$ allerdings auch häufig vorgegeben, da sie durch politische oder wirtschaftliche Rücksichten beeinflußt wird.
Unabhängig von der Wahl der Zielfunktionen ergeben sich bislang die folgenden Restrik-

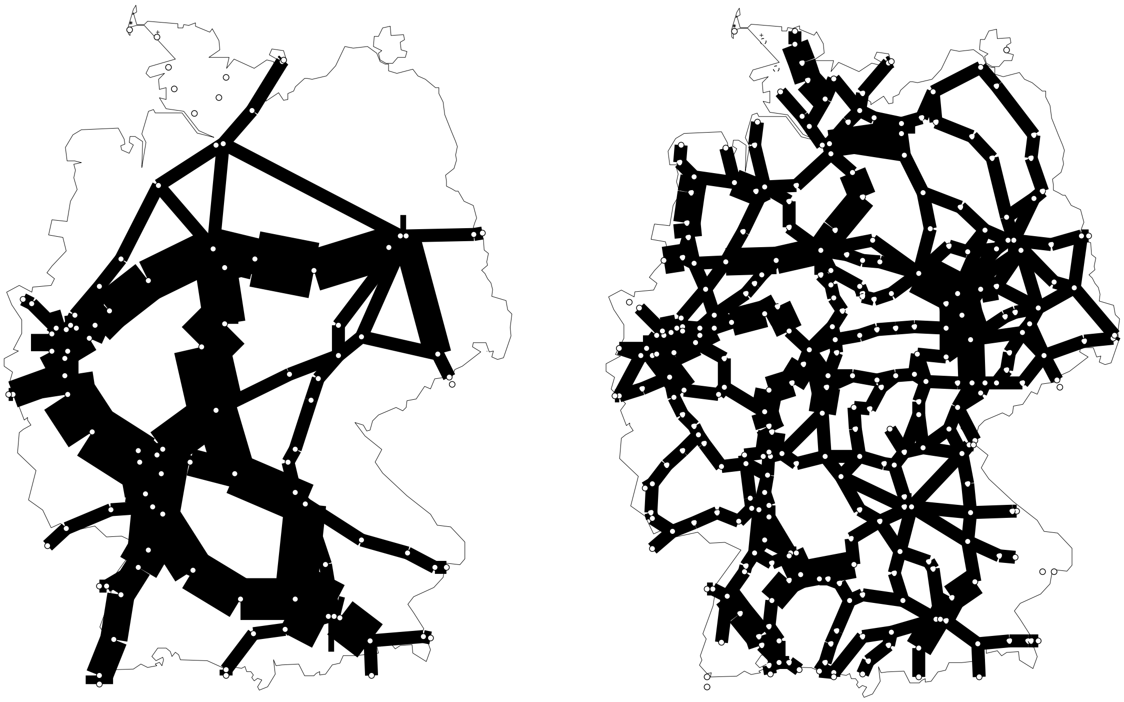


Abbildung II.43: Die Linienbedarfsfrequenz der deutschen IC/ICE- und IR-Netze

tionen:

$$(7.10) \qquad \begin{array}{rcll} \sum_{l \in \mathcal{L}, e \in l} f_l & = & lfr(e) & \\ f & \geq & 0 & (ganz) \end{array}$$

Ein Ziel der Linienoptimierung kann es sein, den Linienplan so zu gestalten, daß möglichst wenig Reisende während ihrer Zugfahrt umsteigen müsen. Bezeichnet man mit $d(t,l)$ die Anzahl der Reisenden, die auf der Linie l zwischen den Stationen a und b ohne Umsteigen reisen können, so ergibt sich als Zielfunktion:

$$(7.11) \qquad z := \max \sum_{(t,l)} d(t,l)$$

Da die Anzahl der Direktreisenden nicht größer als die Gesamtzahl der Reisenden sein kann und die Zugkapazitäten beschränkt sind, ergeben sich zusätzliche Restriktionen:

$$\sum_{l:t\subseteq l} d(t,l) \leq tr(l) \tag{7.12}$$

$$\sum_{t:e\in t\subseteq l} d(t,l) \leq Cf(l) \quad \forall e,l \tag{7.13}$$

Es ergibt sich ein MIP ("mixed integer problem"), welches als **LOP** referiert wird. Da **LOP** derart umfangreich ist, daß die Rechenzeiten unpraktikabel groß werden, oder das Problem gar nicht mehr lösbar ist (vergleiche Tabellen II.1 und II.2), müssen sinnvolle Vereinfachungen vorgenommen werden.

Netz	$\lvert V\rvert$	$\lvert\mathcal{CY}\rvert$	$\lvert E\rvert$	$\lvert\mathcal{L}\rvert$	$\lvert\mathcal{T}\rvert$	**LOP**		
						Variable	Restriktionen	Nonzeros
NS-IC	23	23	31	253	210	2120	1017	6278
NS-IR	86	86	114	3655	2147	116974	28487	556131
NS-CT	385	91	428	4095	11240	583598	96620	3589478
DB-IC	100	100	118	4950	3136	183235	43200	1038761
DB-IR	307	199	398	19701	9215	900173	261308	4853878

Tabelle II.1: Netz-Parameter und MIP-Größe

Netz	Laufzeit		Zielwert		$\lvert\{f_l \mid f_l \notin \boldsymbol{Z}_+\}\rvert$
	LP	MIP	LP	MIP	
NS-IC	10.55	4912.32	9.168.554	8.203.412	141
NS-IR	5284.34	-	21.315.607	-	636
NS-CT	-	-	*25.492.888	-	-
DB-IC	12777.66	-	9.768.973	-	1384
DB-IR	-	-	*8.095.734	-	-

$^{-}$) kein Ergebnis nach 5 h Rechenzeit. Time in seconds. $*$ berechnet von Bixby:95

Tabelle II.2: Berechnungen mit **LOP**

Eine mögliche Vereinfachung besteht aus der Relaxation der Ungleichungen (7.13), die eine Überlastung mit Reisenden der Linien auf jede ihrer Kanten verhindert. Ungleichungen vom Typ (7.13) mit einer Summation über verschiedene Reisepaarungen (bei Transport/Flußproblemen spricht man von Gütern ("commodities") beschränken die Bündelkapazität. Beschränkt man nun nicht die Summe der Reisepaarungen durch $Cf(l)$, sondern jeden einzelnen Summanden durch $Cf(l)$, so ergibt sich folgende Situation, wobei eine Unterscheidung der einzelnen Kanten nicht mehr nötig ist:

$$d(t,l) \leq Cf(l) \quad \forall l \in \mathcal{L},\ \forall t \in \mathcal{T} \tag{7.14}$$

Durch Summation der Ungleichungen in (7.14) ergibt sich

$$\sum_{l:\ t \subset l} d(t,l) \leq C \sum_{l:\ t \subset l} f(l) \tag{7.15}$$

Faßt man nun die Direktreisenden von t in den verschiedenen Linien zu $D(t)t := \sum_{l;\ t \subset l} d(t,l)$ zusammen, so vereinfachen sich die Ungleichungen (7.12) und (7.13) zu:

$$D(t) \leq tr(T) \tag{7.16}$$

$$D(t) \leq C \sum_{t \subseteq l} f(l) \tag{7.17}$$

Das gesamte Problem hat dann die Form:

$$\begin{array}{rrcl} z := \max & \sum_t D(t) & & \\ \text{unter} & Af & = & lfr \\ & D(t) & \leq & C \sum_{t \subset l} f(l) \\ & 0 & \leq & D \leq tr \\ & 0 & \leq & f \ \ ganz \end{array} \tag{7.18}$$

Als Relaxation dieses gemischt ganzzahligen Programmes auf ein lineares Programm ist f schnell bestimmbar, aber im Regelfall natürlich nicht ganzzahlig. Ein Branch & Bound-Verfahren, das mit der LP-Relaxation zur Bestimmung der unteren Schranken arbeitet, erweist sich als zu langsam. Bewährt jedoch hat sich die Idee, zusätzliche Ungleichungen einzuführen, die aus den von $D(t)$ abhängigen Beschränkungen für f abgeleitet werden.

Lemma 7.5 *Sei $0 \leq y \leq a$ und $y \leq bx$. Dann gilt für $\Delta := a - \lfloor \frac{a}{b} \rfloor b$*

$$y \leq \lfloor \frac{a}{b} \rfloor (b-a) + \Delta x = g(x) \quad \forall x \in \mathbb{Z} \tag{7.19}$$

Beweis. siehe Abbildung II.44. □

Aus Lemma 7.5 erhält man für jedes $D(t)$ höchstens eine neue Ungleichung für das lineare Programm.

Lemma 7.6 *Sei $V' \subseteq V$, $E' \subseteq \delta(V')$ und $lfr(E')$ ungerade. Sei weiterhin $\alpha_l := |l \cap E'|$. Dann gilt für jeden ganzzahligen Linienplan f:*

$$\sum_{l:\ \alpha_l\ gerade} \alpha_l f(l) \leq lfr(E') - 1 \tag{7.20}$$

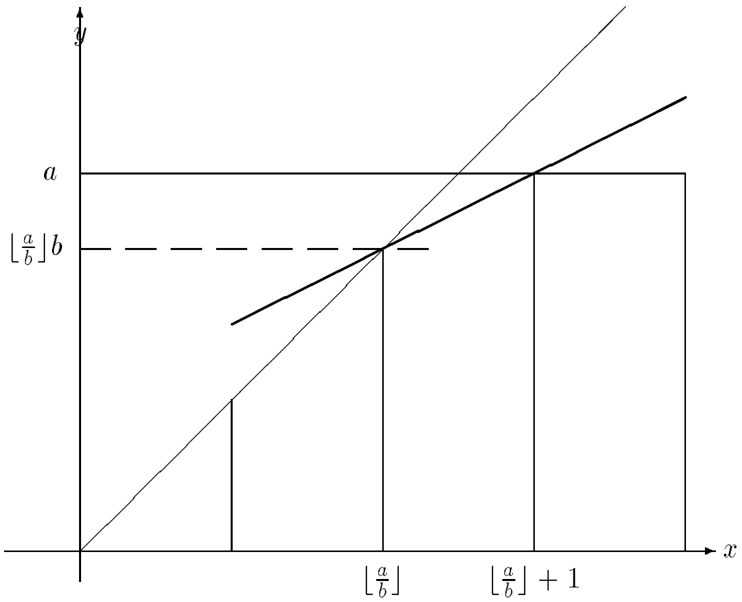


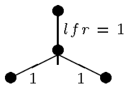
Abbildung II.44: zum Beweis von Lemma 7.5

Beweis.

$$\underbrace{\sum_{\alpha_l l \text{ gerade}} \alpha_l f(l)}_{\text{gerade}} \leq \underbrace{lfr(E')}_{\text{ungerade}}$$

□

Lemma 7.6 liefert zu viele Gleichungen, um sie alle dem linearen Programm hinzuzufügen.

Gleichungen vom Typ 

sind aber höchstens $|\mathcal{CY}|$.
Mit Hilfe dieses MIP, referiert als **lop**, erhält man innerhalb akzeptabler Rechenzeiten sehr gute Ergebnisse (vgl. Tabelle II.3).

8 Rucksackprobleme

(8.1) $\max\{c^T x | a^T x \leq b, x \in \mathbf{Z}_+^n\}$ mit $0 < a_j \leq b, c_j \geq 0$ a, b, c ganz

Beispiel Zuschnittprobleme:

Balken der Länge b, x_j Schnitte der Länge a_j und c_j Wert des Schnittes j

Netze	Größe			Laufzeit		Zielfunktionswert		$\|\{f_l \| f_l \notin \boldsymbol{Z}_+\}\|$
	Variable	Restriktionen	Nonzeros	LP	MIP	LP	MIP	
NS-IC	463	234	3060	0.18	0.66	9.168.554	8.203.412	122
NS-IR	5802	2305	153752	11.07	187.50	28.909.033	21.172.411	502
NS-CT	15335	11763	696630	44.25	2128.47	38.922.436	37.118.270	390
DB-IC	8086	3276	236923	39.54	103.23	10.071.448	7.625.326	512
DB-IR	28915	9692	1239870	94.02	1527.92	8.106.707	6.114.448	1347

Tabelle II.3: Ergebnisse mit **lop**

Rucksackfunktion

(8.2) $$F(k,y) := \max\{\sum_{j=1}^{k} c_j x_j \mid \sum_{j=1}^{k} a_j x_j \leq y, x_1, \cdots, x_k \in \boldsymbol{Z}_+\}$$

für $k = 1, \cdots, n$ $y = 0, 1, \cdots, b$

Bemerkung:

1. $F(k,0) = 0$ für $(1 \leq k \leq n)$
2. $F(1,y) = c_1 \left\lfloor \frac{y}{a_1} \right\rfloor$ für $(0 \leq y \leq b)$
3. $F(n,b)$ ist der Optimalwert von (8.1)

Satz 8.1 *Für $k = 2, 3, \cdots, n$ und $y = 0, 1, \cdots, b$ gilt*

(8.3) $$F(k,y) = \begin{cases} F(k-1,y) & \text{falls } a_k > y \\ \max\{F(k-1,y), c_k + F(k, y-a_k)\} & \text{falls } a_k \leq y \end{cases}$$

Beweis.

1. $F(k-1,y) \leq F(k,y)$, denn x OL zu $(k-1,y) \Rightarrow \begin{pmatrix} x \\ 0 \end{pmatrix}$ zulässig für (k,y).
2. $c_k + F(k, y-a_k) \leq F(k,y)$ falls $a_k \leq y$. x OL zu $(k, y-a_k) \Rightarrow (x_1, \ldots, x_{k-1}, x_k + 1)$ ist zulässig für (k,y).
3. Sei x OL zu (k,y)

   Fall 1 $x_k = 0 \rightsquigarrow (x_1, \ldots, x_{k-1})$ zulässig für $(k-1,y) \overset{1.}{\rightsquigarrow} F(k,y) = F(k-1,y)$.

   Fall 2 $x_k > 0 \rightsquigarrow x_k \geq 1 \rightsquigarrow y \geq a_k \rightsquigarrow (x_1, \ldots, x_{k-1}, x_k - 1)$ zulässig für $(k, y-a_k) \overset{2.}{\rightsquigarrow}$ $c_k + F(k, y-a_k) = F(k,y)$. □

Graphische Darstellung

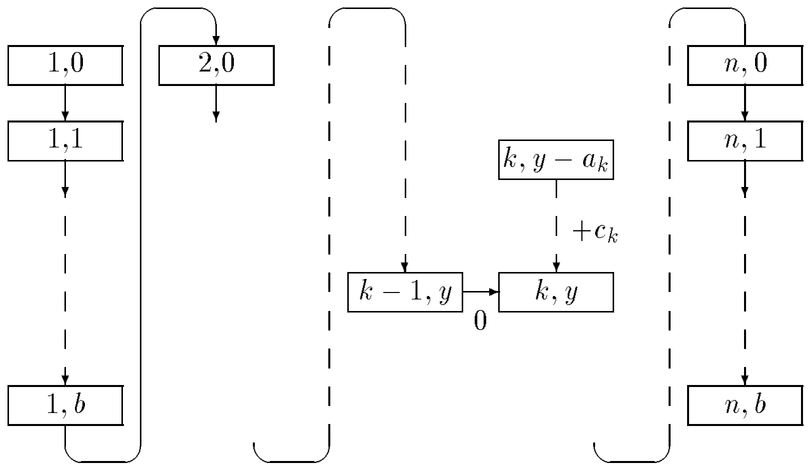


Abbildung II.45: Tableau zur Berechnung der $F(k,y)$

Optimallösung

x OL zu $(k-1,y)$, x' OL zu $(k,y-a_k)$ falls $y \geq a_k$. Dann gilt

$$
\begin{array}{lcl}
F(k,y) = F(k-1,y) & \rightarrow & (x,0) \text{ OL zu } (k,y) \\
F(k,y) > F(k-1,y) & \rightarrow & x' + e_k \text{ OL zu } (k,y)
\end{array}
\tag{8.4}
$$

Speicheraufwand zu groß. Sei $\hat{x}(k,y)$ die so berechnete Lösung.

$$
j(k,y) := \begin{cases} 0 & \hat{x} = 0 \\ \max\{j | \hat{x}_j(k,y) > 0 & \hat{x} \neq 0 \end{cases}
\tag{8.5}
$$

Lemma 8.2 *1.* $j(1,y) = \begin{cases} 1 & a_1 \leq y \leq b \\ 0 & 0 \leq y < a_1 \end{cases}$

2. Für $k = 2,3,\ldots,n$, $y = 0,\ldots,b$ gilt

$$
j(k,y) = \begin{cases} j(k-1,y) & \textit{falls } F(k,y) = F(k-1,y) \\ k & \textit{falls } F(k,y) > F(k-1,y) \end{cases}
\tag{8.6}
$$

Algorithmus 8.1 (Rückwärtsrekursion)

1. $\hat{x} \equiv 0$, $y = b$, $k := j(n,b)$.
2. Falls $k = 0$ stop.
3. $\hat{x}_k := \hat{x}_k + 1$; $y := y - a_k$; $k := j(n,y)$
   gehe nach 2.

Zusammenfassung:

Aufgabe

$$\max\{c^T x \mid a^T x \leq b,\ x \in \boldsymbol{Z}_+^n\}$$

für $a_j \in \boldsymbol{Z}_+$, $(1 \leq j \leq n)$, $b \in \boldsymbol{N}$ nach Gilmore und Gomory.
($a_j \leq b$, $1 \leq j \leq n$ angenommen.)

Anfangsdaten $c_j, a_j (1 \leq j \leq n), b$

1. $x_j := 0\,(1 \leq j \leq n)$

2. $F(1,y) := c_1 \cdot \lfloor y/a_1 \rfloor$ d$(0 \leq y \leq b)$
   $j(1,y) := 0$ $(0 \leq y < a_1)$
   $j(1,y) := 1$ $(a_1 \leq y \leq b)$
   $k := 2$

Optimalwert-Rekursion

1. Für $z \in \{0, 1, \ldots a_k - 1\}$
   $F(k,z) := F(k-1,z)$
   $j(k,z) := j(k-1,z)$
   $y := a_k$.

2. Falls $F(k-1,y) \geq F(k, y - a_k) + c_k$, dann
   $F(k,y) := F(k-1,y)$
   $j(k,y) := j(k-1,y)$
   gehe nach 4.

3. $F(k,y) := F(k, y - a_k) + c_k$
   $j(k,y) := k$.

4. Falls $y < b$, dann
   $y := y + 1$
   gehe nach 2.

5. Falls $k < n$, dann
   $k := k + 1$
   gehe nach 1.

Bestimmung des optimalen Lösungsvektors

1. Setze $y := b$.

2. Falls $j(n,y) = 0$, dann
   STOP (x optimal)

3. $k := j(n,y)$, $x_k := x_k + 1$, $y := y - a_k$
   gehe nach 2.

Beispiel:

Aufgabe

$$\begin{array}{lccccccccccccl} \max & x_1 & + & 2x_2 & + & 3x_3 & + & 4x_4 & + & 6x_5 & & & & \\ \text{unter} & 2x_1 & + & 2x_2 & + & 3x_3 & + & 3x_4 & + & 4x_5 & \leq & 5 & & x \in \mathbf{Z}_+^5 \end{array}$$

Tabelle der $F(k,y)/j(k,y)$:

	x_1	x_2	x_3	x_4	x_5
c_k	1	2	3	4	6
a_k	2	2	3	3	4
$y = 0$	0/0	0/0	0/0	0/0	0/0
$y = 1$	0/0	0/0	0/0	0/0	0/0
$y = 2$	1/1	2/2	2/2	2/2	2/2
$y = 3$	1/1	2/2	3/3	4/4	4/4
$y = 4$	2/1	4/2	4/2	4/2	6/5
$y = 5$	2/1	4/2	5/3	6/4	6/4

$a_2 = 2$

$a_4 = 3$

Lösung: $F(5,5) = 6, x_4 = 1, x_2 = 1$ (sonst 0).
Lösung zu $F(4,4) = 4; x_2 = 2$ (sonst 0).

Komplexität

Länge der Eingabedaten $\alpha = \max\{a_j, c_j, b\}$. $L := O(n \log \alpha)$. Aufwand: $O(nb) \hat{=} O(L2^L)$ exponentiell.

0-1 Rucksackprobleme

(8.7) $$z^* = \max\{c^T x | a^T x \leq b,\ x \in \{0,1\}^n\} \quad \text{mit } c \geq 0, 0 < a_j \leq b$$

Minimale Rucksackgewichte

Für $z = 0, 1, \ldots, \sum_j c_j$, $k = 1, \ldots, n$

(8.8) $$G(k,z) := \begin{cases} \min\{\sum_{j=1}^k a_j x_j | \sum j = 1^k c_j x_j = z;\ x \in \{0,1\}^k\} & \text{falls } \min < b \\ \infty & \text{andernfalls} \end{cases}$$

Bemerkung:

1. $G(k,0) = 0$ für $(1 \leq k \leq n)$

2. $G(1,z) = \begin{cases} a_1 & z = c_1 \\ \infty & z \neq c_1 \end{cases}$ für $0 < z \leq \sum_j c_j$.

3. $z^* = \max\{z | G(n,z) \text{ endlich }\}$

4. $G(n, z^*)$ ist das Gewicht der optimalen Packung von (8.8)

Satz 8.3 *Für $k = 2, \ldots, n$ und $z = 0, 1, \ldots, z^*$ gilt*

$$G(k,z) = \begin{cases} G(k-1,z) & \textit{falls } a_k + G(k-1, z-c_k) > b \\ \min\{G(k-1,z), a_k + G(k-1, z-c_k)\} & \textit{andernfalls} \end{cases}$$

Beweis.

1. $G(k-1,z) \geq G(k,z)$.

2. $a_k + G(k-1, z-c_k) \geq G(k,z)$ falls linke Seite $\leq b$. x OL zu $(k-1, z-c_k) \rightsquigarrow x + e_k$ Lösung zu (k,z).

3. Sei x OL zu (k,z)

   Fall 1 $x_k = 0 \rightsquigarrow (x_1, \ldots, x_{k-1})$ zulässig für $(k-1,z) \overset{1.}{\rightsquigarrow} G(k-1,z) = G(k,z)$.

   Fall 2 $x_k = 1 \rightsquigarrow x - e_k$ zulässig für $(k-1, z-c_k) \rightsquigarrow G(k,z) = a_k + G(k-1, z-c_k)$.

□

Optimallösung

1. $j(1,z) = \begin{cases} 1 & \text{falls } z = c_1 \\ 0 & \text{falls } z \neq c_1 \end{cases} \quad 0 < z \leq \sum c_j$.

2. Für $k = 2, \ldots, n$ gilt $j(k,z) = \begin{cases} j(k-1,z) & \text{falls } G(k-1,z) = G(k,z) \\ k & \text{falls } G(k-1,z) > G(k,z) \end{cases}$

Beispiel:

$$\begin{array}{lccccccccccccc} \max & 6x_1 & + & 11x_2 & + & 17x_3 & + & 3x_4 & + & 9x_5 & & & & \\ \text{unter} & x_1 & + & x_2 & + & 3x_3 & + & 2x_4 & + & 2x_5 & \leq & 5 & & 0 \leq x \leq 1 \end{array}$$

z/k	1	2	3	4	5
0	0/0	0/0	0/0	0/0	0/0
1					
2					
3				3/4	3/4
4					
5					
6	1/1	1/1	1/1	1/1	1/1
7					
8					
9				3/4	2/5
10					
11		1/2	1/2	1/2	1/2
12					5/5
13					
14				3/4	3/4
15					3/5
16					
17		2/2	2/2	2/2	2/2
18					5/5
19					
20				4/4	3/5
21					
22					
23			4/3	4/3	4/3
24					
25					
26					4/5
27					
28			4/3	4/3	4/3
29					
30					
31					
32					
33					
34			5/3	5/3	5/3

Tabelle der $G(k,z)/j(k,z)$

Lösung: $z^* = 34,\ x = (1,1,1,0,0)^T$
Aufwand: $O(n^2 c_{\max})$

9 Approximative Algorithmen

Problem $\min\{c(R) | R$ Rundreise $\}$ $[\mathrm{TSP_{opt}}]$

Instance $n \in \boldsymbol{N}, K_n = (V,E), c : E \to \boldsymbol{Z} \quad I \in [\mathrm{TSP_{opt}}]$

Eingabegröße $< n > + < c >$ mit $< n > := 1 + \lfloor \log_2 |n| + 1 \rfloor$; $< \frac{p}{q} > := < p > + < q >$; p, q teilerfremd.

Entscheidungsproblem Enthält K_n Rundreise R mit $c(R) \leq \lambda$ [TSP]

Eingabegröße von $I \in$ [TSP] $< I > = < n > + < c > + < \lambda >$

Bemerkung: Optimierungsprobleme mit $a \leq \min\{f(x) | x \in P\} \leq b$ mittels binärer Suche rückführbar auf Entscheidungsprobleme. $\quad \exists\, x \in P : f(x) \leq \lambda$.

Entscheidungsproblem π

Gegeben: $I \in \pi$
Frage: Gibt es eine Lösung S zu I?

Polynomiale deterministische Algorithmen

1. Algorithmus $[A]$ löse $I \in \pi$ mittels $f(I)$ elementarer Bitoperationen. Dann heißt $[A]$ polynomial, falls
$$f(I) = O(p(< I >)) \qquad \forall I \in \pi$$
für ein festes Polynom $p : \boldsymbol{N} \to \boldsymbol{N}$.

2. $P := \{\pi \mid$ es gibt einen polynomialen Algorithmus zur Lösung von $\pi\}$.

Polynomiale nichtdeterministische Algorithmen

1. Algorithmus $[B]$ bestehe aus zwei Teilen, einem Ratemodul $[R]$ und einem Testmodul $[T]$, wobei $[R]$ zu $I \in \pi$ eine Lösung S rät und $[T]$ prüft, ob S tatsächlich I löst. Dann heißt $[B]$ polynomial, falls es zwei Polynome $p, q : \boldsymbol{N} \to \boldsymbol{N}$ gibt, für die gilt:
   (a) Falls die Antwort zu I „ja" lautet, so gibt es eine Lösung S mit
      i. $< S > = O(p(< I >))$
      ii. $[T]$ bescheinigt Korrektheit von S für I in $O(q(< I > + < S >))$.
   (b) Falls die Antwort zu I „nein" lautet, so verwirft $[T]$ jede von $[R]$ geratene Lösung S als inkorrekt für I.

2. $NP := \{\pi \mid$ es gibt einen nichtdeterm. polynomialen Algorithmus zur Lösung von $\pi\}$.

Beispiel: [TSP]$\in$ NP denn [R]: Wähle beliebige Rundreise S, $< S >= O(< I >)$. [T]: $c(R) \leq \lambda$ testen in $O(< S > + < I >)$

[GLP$_{\text{opt}}$]: $\min\{c^T x | Ax = b, x \in \boldsymbol{Z}_+^n\}$: [GLP]: [R]: wähle ein $x \in \boldsymbol{Z}_+^n$, [T]: Teste $Ax = b$ und $c^T x \leq \lambda$. Falls ja, so ist bekannt: es gibt ein x mit $< x >$ polynomial in $< I >$. [T] ist polynomial in $< I > + < x > \rightsquigarrow$ [GLP] $\in$ NP.

Bemerkung:

1. P $\subseteq$ NP mit [R]$\equiv$[A], [R] übernimmt Antwort von [R]
2. $\pi \in$ NP $\rightsquigarrow$ existiert deterministischer Algorithmus [A] mit Laufzeit $O(e^{r(<I>)})$ für ein Polynom r. Alle S mit $< S >= O(p(< I >))$. Diese exponentiell vielen $(O(e^{r(<I>)}))$ mit [T] testen.

Polynomiale Transformationen $\pi \stackrel{\text{poly}}{\longrightarrow} \pi'$

1. $I \rightarrow I'$ mit polynomialem deterministischen Algorithmus [A]
2. I lösbar $\Leftrightarrow$ I' lösbar.
3. Polynomiale Transformationen sind transitiv.

NP-vollständige Probleme

$\pi \in$ NP mit $\pi' \stackrel{\text{poly}}{\longrightarrow} \pi$ für alle $\pi' \in$ NP

Beispiel: [0-1-GLP]: Gibt es ein $x \in \{0, 1\}^n$ mit $Ax \leq b$

[SAT] Satisfiability

Boolesche Operationen: $+, \cdot$ auf $\{0, 1\}$ mit $1+1 = 1$, Komplementoperationen $\bar{1} = 0, \bar{0} = 1$.

Gegeben: $C_\mu := \sum_{i \in I_\mu} x_i + \sum_{i \in K_\mu} \bar{x}_i$ für $\mu = 1, \ldots, m$, $i \in \{1, \ldots, n\}$ (also $T_\mu, K_\mu \subseteq \{1, \ldots, n\}$.

Frage: $\quad \exists\, x \in \{0, 1\}^n : \underbrace{\prod_{\mu=1}^{m} C_\mu(x)}_{=: F(x)} = 1$

Satz 9.1 (Cook,1971) *[SAT] ist NP-vollständig.*

Beweis. $5\frac{1}{2}$ Seiten in Garey/Johnson.

Leicht: [SAT]$\in$ NP, denn [R]: Wähle $x \in \{0, 1\}^n$, [T]: Berechne $F(x)$.

Schwer: "vollständig". □

Korollar 9.2 *[0-1-GLP] ist NP-vollständig.*

Beweis.

1. [0-1-GLP]$\in$ NP: [R]: Wähle $x \in \{0,1\}^n$, [T]: Teste $Ax \leq b$.

2. für $0-1$ Lösungen gilt:

$$\begin{aligned} F(x) = 1 \quad &\Leftrightarrow \quad \sum_{i \in I_\mu} x_i + \sum_{i \in K_\mu} \bar{x}_i = 1 \quad \forall\, \mu \quad \text{Boolesche Operationen} \\ &\Leftrightarrow \quad \sum_{i \in I_\mu} x_i + \sum_{i \in K_\mu} (1 - x_i) \geq 1 \quad \forall\, \mu \quad \text{ganzzahlige Operationen} \\ &\Leftrightarrow \quad Ax \leq b \end{aligned}$$

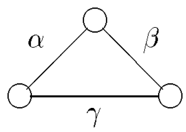
$\square$

Bemerkung:

1. Es gibt sehr viele NP-vollständige Probleme (siehe Garey/Johnson)

2. [TSP] ist NP-vollständig (siehe Papadimitriou/Steiglitz)

3. [$\triangle$-TSP] ist NP-vollständig. Zusätzlich zu [TSP] erfülle c die $\triangle$-Ungleichung.



$$\longrightarrow \quad \alpha + \beta \geq \gamma \geq 0$$

Abbildung II.46: Dreicksungleichung der Gewichtsfunktion

4. Hypothese: P $\subset$ NP.

5. [HAM] ist NP-vollständig: Enthält $G = (V, E)$ eine Rundreise.

Algorithmus 9.1 (Baumheuristik für [$\triangle$-TSP])

1. Bestimme minimales Gerüst $\hat{B}$ von K_n.

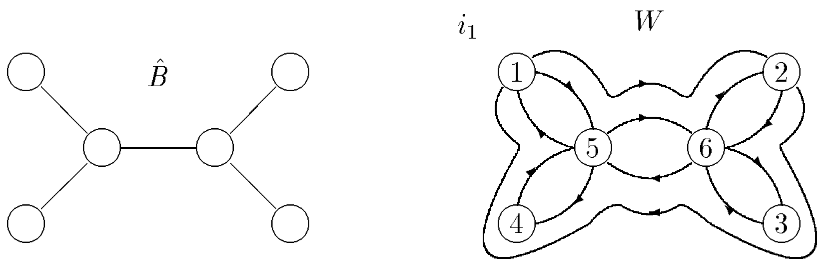
2. Verdopple Kanten von $\hat{B}$; Ergebnis sei $\hat{E}$.

3. Bestimme Eulerschen Weg W in $\hat{E}$; beginnend bei i_1. Sei seine Knotenreihenfolge $(i_1, i_2, \ldots, i_n) =: R$.

Güte der Heuristik

$\hat{c} \hat{=} c(\hat{R})$, $\hat{R}$ optimale Rundreise.

$$c(R) \leq 2c(\hat{B}) \leq 2c(\hat{R}) \rightsquigarrow c(R) \leq 2c(\hat{R}) \rightsquigarrow \frac{c(R) - c(\hat{R})}{c(\hat{R})} \leq 1.$$



$(1, 5, 6, 2, 3, 4) = R$
Aufwand: $O(n^2)$

Abbildung II.47: Baumheuristik

ϵ-approximativ

$\hat{c}$ Minimum. Algorithmus [A] liefere Näherungslösung c. [A] heißt ϵ-approximativ, falls

$$\frac{c - \hat{c}}{\hat{c}} \leq \epsilon$$

für alle $I \in \pi$.

Polynomiale Approximationsschemata (PAS)

Algorithmus $[A]$ bestimme eine ϵ-approximative Lösung zu $I \in \pi_{\text{opt}}$ (Optimierungsproblem) in der Laufzeit $f(I, \epsilon)$.

1. $[A]$ heißt **polynomiales Approximationsschema**, falls für alle rationalen $\epsilon > 0$ ein Polynom $p_\epsilon : \boldsymbol{N} \to \boldsymbol{N}$ existiert mit $f(I, \epsilon) = O(p_\epsilon(< I >))$.

2. Ein (PAS) heißt vollständig, falls ein Polynom $p : \boldsymbol{N}^2 \to \boldsymbol{N}$ existiert mit $f(I, \epsilon) = O(p(< \frac{1}{\epsilon} >, < I >))$.

Perfekte Matchings

$G = (V, E)$ ungerichtet, $|V|$ gerade. $M \subseteq E$ heißt perfektes Matching, falls

$$x_M(\delta(i)) = 1 \quad \forall\, i \in V$$

$\min\{c(M)|M \text{ perfektes Matching }\}$ in $O(n^3)$.

Algorithmus 9.2 (Baum/Matching-Heuristik (Christofides))

1. Bestimme minimales Gerüst $\hat{B}$ von K_n.

2. (a) $V' := \{i | \delta(i) \cap \hat{B} \text{ ungerade }\}$

   (b) Sei M' minimales perfektes Matching in $(V', E(V'))$

   (c) $\hat{E} := \hat{B} \cup M'$

3. Bestimme Eulerschen Weg W in $\hat{E}$; beginnend bei i_1. Sei seine Knotenreihenfolge $(i_1, i_2, \ldots, i_n) =: R$.

Güte der neuen Heuristik

$\epsilon = \frac{1}{2}$.
Beweis. $|V'|$ gerade, $(V', \hat{B} \cup M')$ eulersch.

1. $c(R) \leq c(\hat{B}) + c(M')$ (Dreiecksungleichung). $c(\hat{B}) \leq \hat{c} = c(\hat{R})$ ($\hat{R} \setminus e$ Baum für $e \in \hat{R}$)

2. $\hat{R}$ durchlaufe V' in Reihenfolge $j_1, j_2, \ldots, j_{2k}$ (vgl. Abbildung II.48.

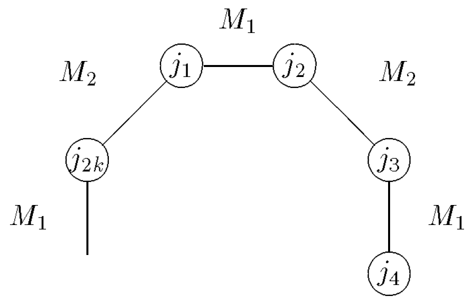


Abbildung II.48: zum Beweis

$\leadsto 2c(M') \overset{M' \text{ min}}{\leq} c(M_1) + c(M_2) \overset{\triangle}{\leq} c(\hat{R}) = \hat{c} \overset{1.+2.}{\leadsto} c(R) \leq c(\hat{B}) + c(M') \leq \hat{c} + \frac{1}{2}\hat{c} \leadsto \frac{c(R)-\hat{c}}{\hat{c}} \leq \frac{1}{2}$ □

Satz 9.3 *Falls $P \neq NP$, so gibt es für kein $\epsilon > 0$ einen polynomialen ϵ-approximativen Algorithmus zu [TSP].*

Beweis. Sei $G = (V, E)$. [HAM] ist NP-vollständig. $|V| = n$. Sei $G' = K_n$.

$$c(e) := \begin{cases} 1 & e \in E \\ 2 + \epsilon n & e \notin E \end{cases}$$

zu zeigen: ϵ-approximativer Algorithmus liefert R mit $c(R) = n \Leftrightarrow G$ enthält Rundreise.

1. $c(R) = n \leadsto R \subseteq E \leadsto G$ enthält Rundreise.

2. G enthalte Rundreise $\hat{R} \leadsto \hat{R}$ optimal. Annahme $c(R) > n \leadsto c(R) \geq (n-1) + (2 + \epsilon n) = n + 1 + \epsilon n \leadsto \frac{c(R)-c(\hat{R})}{c(\hat{R})} = \frac{1+\epsilon n}{n} = \frac{1}{n} + \epsilon > \epsilon$ Widerspruch. □

9.1 0-1 Rucksackprobleme

$$z_* = \max\{c^T x | a^T x \leq b, x \in \{0,1\}^n\} \quad c \geq 0, 0 > a_j \leq b \quad \forall\, j, c_{\max} = \max_j c_j$$

ist NP-vollständig und lösbar in $O(n^2 c_{\max})$.

Skalierung der Kosten

$c_j \longrightarrow c_j(t) := \lfloor \frac{c_j}{2^t} \rfloor \, 2^t \rightsquigarrow c_j - 2^T \leq c_j(t) \leq c_j.$

Fehler bei Lösung des skalierten Problemes

$x(t)$ OL zu $c(t)$, x OL zu c.

$$\begin{aligned} c^T x &\geq c^T x(t) \geq c(t)^T x(t) \geq c(t)^T x \geq \sum_{x_j=1} (c_j - 2^t) \geq c^T x - n2^t \\ n2^T &\geq c^T x - c^T x(t) \geq 0 \\ \epsilon := \frac{n2^t}{c_{\max}} &\geq \frac{c^T x - c^T x(t)}{c^T x} \quad (\text{da } c_{\max} \leq c^T x) \quad t = t(\epsilon) = \lfloor \log \frac{\epsilon c_{\max}}{n} \rfloor \end{aligned}$$

Aufwand bei Lösung des skalierten Problems

Eingabe: $\tilde{c}(t) := \frac{c(t)}{2^t} \rightsquigarrow O(n^2 \frac{c_{\max}}{2^t}) = O(n^3 \frac{1}{\epsilon})$ d.h. Lösung zu $t = t(\epsilon) = \lfloor \log \frac{\epsilon c_{\max}}{n} \rfloor$ liefert VPAS.

Beispiel:

<u>**Aufgabe**</u>

$$\begin{aligned} \max \quad & 6x_1 + 11x_2 + 17x_3 + 3x_4 + 9x_5 \\ \text{unter} \quad & x_1 + x_2 + 3x_3 + 2x_4 + 2x_5 \leq 5 \\ & x \in \{0,1\}^5 \end{aligned}$$

Lösung: $x^* = (1,1,1,0,0)^T$ mit $z^* = 34$.

Gütegarantie: $z^* - \frac{\hat{z}}{z^*} \leq \epsilon$ bei Streichen von $t = \lfloor \log \frac{17}{5} \epsilon \rfloor$ binären Stellen.

z.B. $\epsilon = \frac{20}{17} \quad \Rightarrow \quad t = 2$

$\Rightarrow \quad \max\{x_1 + 2x_2 + 4x_3 + 2x_5 \mid x_1 + x_2 + 3x_3 + 2x_4 + 2x_5 \leq 5,\ x \in \{0,1\}^5\}$

ϵ	t	$z^* - \frac{\hat{z}}{z^*}$	Lösung
10/17	1	0	x^*
20/17	2	0	x^*
40/17	3	8/34	(0 0 1 0 1)
80/17	4	1/2	(0 0 1 0 0)

10 Relaxation und Dualität

$$(10.1) \quad z(b) := \max\{c^T x \mid \underbrace{Ax \leq b, x \in \boldsymbol{Z}^n_+}_{P_g(b)}\} \leq z_{LP}(b) := \max\{c^T x \mid \underbrace{Ax \leq b, x \geq 0}_{P(b)}\}$$

mit $b \in S \subseteq \boldsymbol{R}^m$.

Bemerkung:

1. keine zulässige Lösung: $\max\{c^T x | Ax \leq b, x \in \boldsymbol{Z}^n_+\} := -\infty$
   $c^T x$ unbeschränkt: $\max\{c^T x | Ax \leq b, x \in \boldsymbol{Z}^n_+\} := \infty$

2. Aus $b \leq b'$ folgt $z(b) \leq z(b')$ und $z_{LP}(b) \leq z_{LP}(b')$, z und z_{LP} sind also nichtfallend.

Lemma 10.1 *1.* $0 \leq z(0) = z_{LP}(0)$

2. $z(0) = 0 \iff D := \{y | y^T A \geq c^T, y \geq 0\} \neq \emptyset$

3. $z(0) = \infty \iff D = \emptyset$

Beweis. $0 \leq z(0) \leq z_{LP}(0)$.
Für $D \neq \emptyset$ ist das LP beschränkt und daher $z_{LP}(0) = 0$, also $z(0) = 0$.
Sei $D = \emptyset$. Aus $P(0) \neq \emptyset$ folgt, daß $c^T x$ unbeschränkt auf rationaler Extremalrichtung h ist. Wählt man h o.B.d.A. ganz, so folgt, daß $c^T x$ auch unbeschränkt auf $P_g(0)$ ist, so daß daher $z(0) = \infty$ gilt. □

Korollar 10.2 *1.* $z(0) = 0 \iff z(b) < \infty \quad \forall b \in \boldsymbol{R}^m$

2. $z(0) = \infty \iff z(b) = \pm\infty \quad \forall b \in \boldsymbol{R}^m$

O.B.d.A. kann also $D \neq \emptyset$ vorausgesetzt werden, da sonst nur noch die Existenz zulässiger Punkte zu charakterisieren bleibt.

Superadditivität

Eine Abbildung $z : S \to \boldsymbol{R}$ heißt superadditiv auf S, falls

$$z(b) + z(b') \leq z(b + b')$$

für alle $b, b' \in S$ mit $b + b' \in S$ gilt.

Lemma 10.3 *Sei $D \neq \emptyset$. Dann gilt: z ist nichtfallend und superadditiv; $z(0) = 0$.*

Beweis. Zu zeigen ist nur noch die Superadditivität.

1. Fall: Seien $x \in P_g(b)$ und $x' \in P_g(b')$. Dann gilt auch $x + x' \in P_g(b + b')$.
   Mit $c^T x = z(b)$ und $c^T x' = z(b')$ folgt $z(b) + z(b') = c^T x + c^T x' = c^t(x + x') \leq z(b + b')$

2. Fall: Sei nun $P_g(b) = \emptyset$ oder $P_g(b') = \emptyset$. Dann gilt $z(b) + z(b') = -\infty \leq z(b + b')$

□

Superadditive Dualität

$$\mathcal{F} := \{F : \boldsymbol{R}^m \longrightarrow \boldsymbol{R} \cup \{-\infty\} | F \text{ nichtfallend, superadditiv}, F(0) = 0\}$$

$$t := \min\{F(b) | \underbrace{F \in \mathcal{F} \text{ mit } F(A_j) \geq c_j,\ j = 1, \ldots, n}_{D_g}\}$$

Bemerkung: Sei $y \in D \neq \emptyset$, also $y^T A \geq c^T$ und $y \geq 0$. Mit $F(b) := y^T b$ folgt $F \in D_g$ und damit $t \leq \min\{y^T b | y^T A \geq c^T, y \geq 0\}$

Lemma 10.4 *(schwache Dualität) Für $F \in D_g$ und $x \in P_g(b)$ gilt:*

$$c^T x \leq F(b)$$

.

Beweis. $c^T x \leq \sum_{j=1}^n F(A_j) x_j \leq F(Ax) \leq F(b)$, da F superadditiv, $F(0) = 0$, x ganz und F nichtfallend ist. □

Korollar 10.5 *Für $P_g(b) \neq \emptyset$ und $z(b) = \infty$ gilt: $D_g = \emptyset = D$.*

Beweis. Aus $z(b) = \infty$ folgt $z(0) = \infty$. Es existiert also ein $x \in P_g(0)$ mit $1 < c^T x$. Aus der Existenz eines $F \in D_g$ würde $1 < c^T x \leq F(0)$ folgen, ein Widerspruch. □

Satz 10.6 *(starke Dualität) Sei $D \neq \emptyset$. Dann gilt: $z(b) = t$*
(Für $P_g(b) = \emptyset$ gilt $t = -\infty$)

Beweis. Für die Optimalwertfunktion z gilt wegen $D \neq \emptyset$: $z(0) = 0$ und $z(A_j) \geq c_j$, $j = 1, \ldots, n$ ($e_j \in P_g(A_j) \rightsquigarrow z(A_j) \geq c_j$). Daher ist $z \in D_g$ und demnach $t \leq z(b)$. Zusammen mit der schwachen Dualität folgt $z(b) = t$. □

Bemerkung: Man kann endlich viele $F_i \in \mathcal{F}, F_i : \boldsymbol{R}^m \longrightarrow \boldsymbol{R}, i = 1, \ldots, q$ angeben, so daß

$$z(b) = \min F_i(b)$$

für alle b mit $z(b) > -\infty$ gilt.

Beispiel:

Für $b \in \boldsymbol{R}$:

$$\begin{array}{rccccccccl} z(b) = \max & 3x_1 & + & 6x_2 & + & 11x_3 & + & 12x_4 & & \\ \text{unter} & x_1 & + & 2x_2 & + & 3x_3 & + & 4x_4 & \leq & b \\ & & & & & & & x & \in & \boldsymbol{Z}_+^4 \end{array}$$

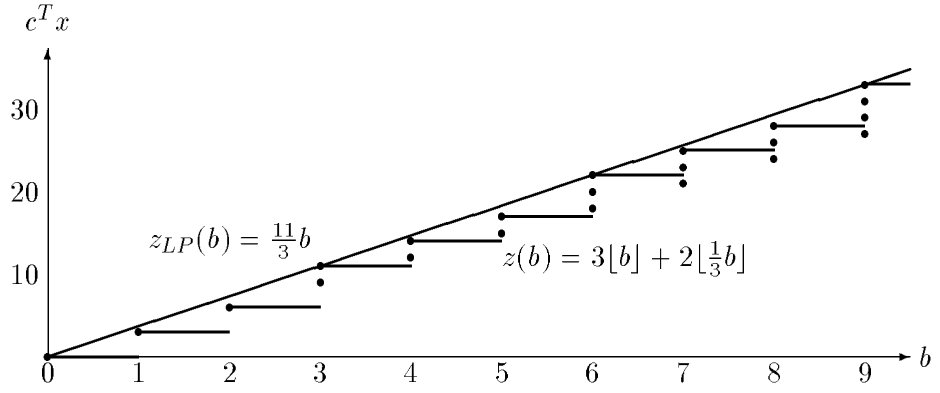


Abbildung II.49: zum Beispiel: zulässige Punkte und Optimalwertfunktionen

Das LP-duale Optimum ist $y = \frac{11}{3}$.

Die LP-Optimalwertfunktion ist $z_{LP}(b) = \frac{11}{3}b$.

Die ganzzahlige Optimalwertfunktion lautet $z(b) = \begin{cases} 3\lfloor b \rfloor + 2\lfloor \frac{1}{3}b \rfloor & b \geq 0 \\ -\infty & b < 0 \end{cases}$

Für $b = 3d$ mit $d \in \boldsymbol{Z}_+$ gilt: $z_{LP}(3d) = 11d = 3(3d) + 2d = z(3d)$.

Im folgenden Abschnitt über die Lagrangesche Relation bleibt b fest. Statt $z(b)$ oder $P_g(b)$ wird daher verkürzend auch z und P_g geschrieben.

Lagrangesche Relaxation LR(λ)

Bei der Lagrangeschen Relaxation werden die Restriktionen des GLP in zwei Gruppen von Restriktionen unterschiedlichen Schwierigkeitsgrades zerlegt:

$$\begin{array}{rcll} A^1 x & \leq & b^1 & \text{(schwierig)} \\ A^2 x & \leq & b^2 & \text{(einfach)} \\ x & \in & \boldsymbol{Z}_+^n & \end{array} \tag{10.2}$$

Dann ist:

$$\begin{array}{rl} z_{LR}(\lambda) := \max & \underbrace{c^T x + \lambda^T (b^1 - A^1 x)}_{z_{LR}(x,\lambda)} \\ \text{unter} & \left.\begin{array}{rcl} A^2 x & \leq & b^2 \\ x & \in & \boldsymbol{Z}_+^n \end{array}\right\} Q \end{array} \tag{10.3}$$

für $\lambda \in \boldsymbol{R}_+^{m_1}$, wobei $z_{LR}(x, \lambda)$ linear in x ist.

Lemma 10.7 $z_{LR}(\lambda) \geq z$

Beweis. Aus $x \in P_g$ folgt, x ist zulässig für LR(λ). Daher gilt $x \in Q$ und $b^1 - A^1 x \geq 0$, also $c^T x \leq c^T x + \lambda^T(b^1 - A^1\lambda)$. □

Lagrangesche Dualität

(10.4)
$$z \leq z_{LD} := \min_{\lambda \geq 0} z_{LR}(\lambda)$$

Satz 10.8 $z_{LD} = \max\{c^t x | A^1 x \leq b^1, x \in \text{conv } Q\}$

Beweis. Aus
$$z_{LR}(\lambda) = \max_{x \in Q} z_{LR}(x, \lambda) = \max_{x \in \text{conv } Q} z_{LR}(x, \lambda)$$
folgt
$$z_{LD} = \min_{\lambda \geq 0} \max_{x \in \text{conv } Q} [c^T x + \lambda(b^1 - A^1 x)].$$
Falls $Q = \emptyset$ ist, so gilt $\max\{c^t x | A^1 x \leq b^1, x \in \text{conv } Q\} = -\infty = z_{LD}$.
Sei conv Q = conv G + conv H. Dann gilt:
$$\max_{x \in \text{conv } Q} [c^T x + \lambda(b^1 - A^1 x)] = \begin{cases} \infty & \text{falls } 0 < (c^T - \lambda^T A^1)H_j \text{ für ein } j \\ c^T G_k + \lambda^t(b^1 - A^1 G_k) & \text{sonst} \end{cases}$$

Daraus ergibt sich:
$$\begin{array}{rllll} z_{LD} = \min & \eta & & & \\ \text{unter} & \eta & +\lambda^t(A^1 G_k - b^1) & \geq & c^T G_k \quad \forall k \\ & & \lambda^T A^1 H_j & \geq & c^T H_j \quad \forall j \\ & & \lambda & \geq & 0 \end{array}$$

LP-Dualisierung führt auf:
$$\begin{array}{rrcll} z_{LD} = max & c^T(\sum \alpha_k G_k + \sum \beta_j H_j) & & & \\ unter & \sum \alpha_k & = & 1 & \\ & A^1(\sum \alpha_k G_k + \sum \beta_j H_j) & \leq & b^1(\sum \alpha_k) & \\ & \alpha_k & \geq & 0 & \forall k \\ & \beta_j & \geq & 0 & \forall j \end{array}$$

Mittels der Dekompositionstechnik von Dantzig und Wolfe erhält man:
$$\begin{array}{rrcl} z_{LD} = max & c^T x & & \\ unter & A^1 x & \leq & b^1 \\ & x & \in & \text{conv } Q \end{array}$$

□

Korollar 10.9 *1. z_{LD} kann (theoretisch) aus den LP's im Beweis berechnet werden.*

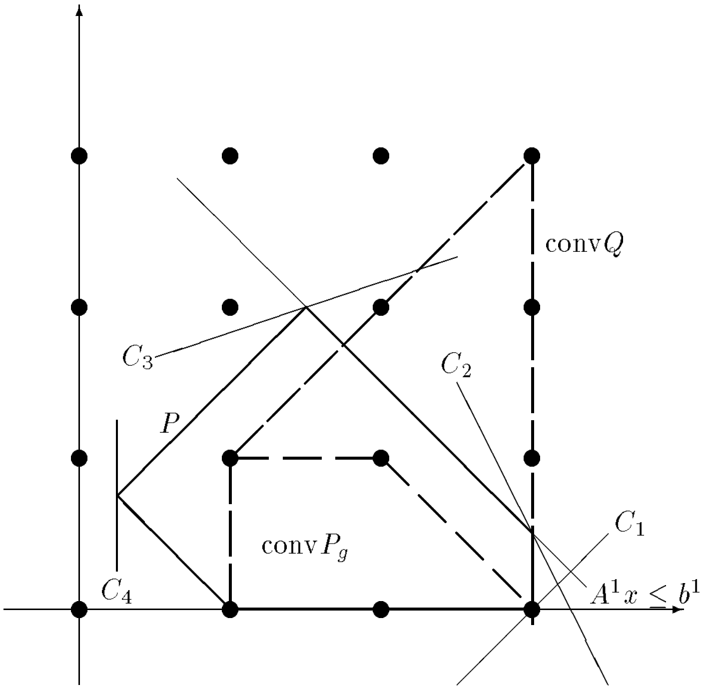
2. $z_{LR}(\lambda)$ ist stückweise linear und konvex, wo endlich (als Optimalwertfunktion mit Parameter in Zielfunktion).

3. $z_{LD} = z$ für alle $c \Longleftrightarrow \mathrm{conv}\,(\mathrm{P_g}) = \mathrm{conv}\ \mathrm{Q} \cap \{\mathrm{x}|\mathrm{A}^1\mathrm{x} \leq \mathrm{b}^1, \mathrm{x} \geq 0\}$

4. $z \leq z_{LD} \leq z_{LP}$

5. $z_{LD} = z_{LP}$ für alle c, falls $\{x|A^2x \leq b^2, x \geq 0\}$ ganz.

Beispiel:



C_1	:	z	$=$	z_{LD}	$=$	z_{LP}
C_2	:	z	$<$	z_{LD}	$=$	z_{LP}
C_3	:	z	$<$	z_{LD}	$<$	z_{LP}
C_4	:	z	$=$	z_{LD}	$<$	z_{LP}

Abbildung II.50: Lagrange-Dualität

Satz 10.10 *$z \geq z_{LD} - \epsilon$ gilt genau dann, wenn $\lambda_* \geq 0$, $x_* \in P_g$ mit $\lambda_*^T(b^1 - A^1 x_*) \leq \delta_1$, $z(\lambda_*, z_*) \geq z_{LR}(\lambda_*) - \delta_2$ und $\delta_1 + \delta_2 \leq \epsilon$ existieren.*

Beweis.

"$\Longleftarrow$" $z_{LD} \leq z_{LR}(\lambda_*) \leq z(\lambda_*, x_*) + \delta_2 \leq c^T x_* + \delta_1 + \delta_2 \leq z + \epsilon$

"$\Longrightarrow$" Sei $x_* \in P_g$ optimal, $z = c^T x_*$, λ_* optimal und $z_{LD} = z_{LR}(\lambda_*)$. Dann gilt:

$$\begin{aligned} z_{LD} &= z_{LR}(\lambda_*) = z(\lambda_*, x_*) + z_{LR}(\lambda_*) - z(\lambda *, x_*) \\ &= \underbrace{c^T x_*}_{z} + \underbrace{\lambda_*^T(b^1 - A^1 x_*) + z_{LR}(\lambda_*) - z(\lambda_*, x_*)}_{\leq \epsilon} \end{aligned}$$

Wähle also $\delta_1 := \lambda_*^T(b^1 - A^1 x_*)$ und $\delta_2 := z_{LR}(\lambda_*) + z(\lambda_*, x_*)$.

□

Korollar 10.11 ($\delta_1 = \delta_2 = \epsilon = 0$)
$z = z_{LD}$ gilt genau dann, wenn $\lambda_* \geq 0$, $x_* \in P_g$ mit $\lambda_*^T(b^1 - A^1 x_*) = 0$ und $z_{LR}(\lambda_*) = z(\lambda_*, x_*)$ existieren.

Beispiel: (Zuordnungsprobleme mit Budget-Beschränkung)(NP-vollständig)

$$\begin{array}{lllll} \max & \sum c_{ij} x_{ij} & & & \\ \text{unter} & \sum_j x_{ij} & = & 1 & \forall i \quad (1) \\ & \sum_i x_{ij} & = & 1 & \forall j \quad (2) \\ & \sum t_{ij} x_{ij} & \leq & b & \quad\ \ (3) \\ & x & \in & \{0,1\}^{n^2} & \end{array}$$

1. Wählt man die Restriktion (3) in die Zielfunktion, ergibt sich:

$$\max\{\lambda b + \sum_{ij} (c_{ij} - \lambda t_{ij}) x_{ij} | x \text{ Matching}\}$$

2. Wählt man die Restriktionen (1) und (2) in die Zielfunktion, so erhält man:

$$\max\{\underbrace{\sum u_i + \sum v_j + \sum_{ij} (c_{ij} - u_i - v_j) x_{ij}}_{z_{LR}(u,v)} \,|\, \underbrace{\sum_{ij} t_{ij} x_{ij} \leq b, x \in \{0,1\}^{n^2}}_{\text{Rucksackproblem}}\}$$

3. Wählt man eine der Restriktionen (1) oder (2) in die Zielfunktion, so ergibt sich ein Rucksackproblem mit verallgemeinerten oberen Schranken.

4. Wählt man eine der beiden Restriktionen (1) oder (2) und zusätzlich die Restriktion (3) für die neue Zielfunktion, ergibt sich (mit (1))

$$\max\{\underbrace{\lambda b+\sum u_i+\sum_{ij}(c_{ij}-u_i-\lambda t_{ij})x_{ij}}_{z_{LR}(u,\lambda)}\,|\,\underbrace{\sum_i x_{ij}=1, ganz}_{\text{"trivial"}}\}$$

(zu j wähle $x_{ij}=1$. Das ist gleichwertig damit, daß $c_{ij}-u_i-\lambda t_{ij}$ maximal ist.)

Bemerkung:

1. 1. und 4. sind ganz Polytope: $z^1_{LD}=z^4_{LD}=z_{LP}$
2. $Q^3\subseteq Q^2$. Also: $z\le z^3_{LD}\le z^2_{LD}\le z^1_{LD}=z^4_{LD}=z_{LP}$.

Verschärfung der Schranken

Das GLP kann umformuliert werden zu:

(10.5)
$$\begin{array}{rlllll} z=\max & c^Tx^1 & & & & \\ \text{unter} & x^1-x^2 & = & 0 & & \text{"schwierig"} \\ & A^1x^1 & \le & b^1 & x^1\in \boldsymbol{Z}^n_+ & (Q_1) \\ & A^2x^2 & \le & b^2 & x^2\in \boldsymbol{Z}^n_+ & (Q_2) \end{array}$$

Lagrangesche Relaxation mit u (Kostenaufspaltung)

(10.6)
$$\begin{array}{rlllll} z'_{LD} & := & \min\limits_u \max & [(c-u)^Tx^1 & + \quad u^Tx^2] & \\ & & \text{unter} & A^1x^1 & & \le b^1 \\ & & & & A^2x^2 & \le b^2 \\ & & & & x^1,x^2 & \in \boldsymbol{Z}^n_+ \\ & = & \min\limits_{c_1+c_2=c} & [\max c_1^Tx^1 & + \quad \max c_2^Tx^2] & \\ & & \text{unter} & A^1x^1 & & \le b^1 \\ & & & & A^2x^2 & \le b^2 \\ & & & & x^1,x^2 & \in \boldsymbol{Z}^n_+ \end{array}$$

Außerdem liefert das umformulierte GLP nach Satz 10.8:

(10.7)
$$\begin{aligned} z'_{LD} &:= \max\{c^Tx^1|x^1-x^2=0, x^1\in \text{conv } Q_1, x^2\in \text{conv } Q_2\} \\ &= \max\{c^Tx^1|x\in \text{conv } Q_1\cap \text{conv } Q_2\}\le_{LD} \end{aligned}$$

Bemerkung:

1. Für einige c gilt:
   $z'_{LD}<z_{LD}$, falls conv $Q_1\subset\{x|A^1x\le b^1, c\in \boldsymbol{R}^n_+\}$.

2. Nützlich, falls separate Beschränkungen $A^i x \leq b^i$, $i = 1, 2$ "einfacher" zu behandeln sind.

3. im Budget-Beipiel:

$$Q_1 : \begin{cases} \sum_j x_{ij} &= 1 \ \forall i \\ \sum t_{ij} x_{ij} &\leq b \\ x &\in \{0,1\}^{n^2} \end{cases} \qquad Q_2 : \begin{cases} \sum_i x_{ij} &= 1 \ \forall j \\ \sum t_{ij} x_{ij} &\leq b \\ x &\in \{0,1\}^{n^2} \end{cases}$$

$$z'_{LD} \leq z^3_{LD} := \max\{\sum c_{ij} x_{ij} | \sum_j x_{ij} = 1 \ \ \forall i, x \in \text{conv } Q_2\}$$

Surrogate Dual (SD)

Sei $\lambda \geq 0$.

$$\begin{aligned} z_{SD}(\lambda) &:= \max\{c^T x | \lambda^T A^1 x \leq \lambda^T b^1, x \in Q\} \\ &\leq \max\{c^T x + \lambda^T (b^1 - A^1 x) | x \in Q\} \\ &= z_{LR}(\lambda) \end{aligned} \tag{10.8}$$

Bemerkung: Werden alle Restriktionen relaxiert, erhält man ein Rucksackproblem:

$$z_{SD}(\lambda) = \max\{c^T x | \lambda^T A x \leq \lambda^T b, \underbrace{x \in \text{conv } \boldsymbol{Z}^n_+}_{x \geq 0}\} \tag{10.9}$$

Benders Dekomposition

$$\begin{aligned} z = \max \quad & c^T x + d^T y \\ \text{unter} \quad & Ax + Dy \leq b \\ & x \in X \subseteq Z^n_+, y \geq 0 \end{aligned} \tag{10.10}$$

(schwierige Variablen $x \in X$.)

$$z_{LP}(x) = \max\{d^T y | Dy \leq b - Ax, y \geq 0, x \in X\} \tag{10.11}$$

$$z_{LP} = \min\{u^T (b - Ax) | \underbrace{u^T D \geq d^T, u \geq 0}_{D(d)}, x \in X\} \tag{10.12}$$

Representation von $D(d)$

$D(0) = \text{cone } H$
Für $D(d) \neq \emptyset$ gilt: $D(d) = \text{conv } G + \text{cone } H$

Lemma 10.12 *1.* $D(d) = \emptyset \iff z_{LP}(x) = \begin{cases} +\infty & \textit{falls } H^T(b - Ax) \geq 0 \\ -\infty & \textit{sonst} \end{cases}$

2. $D(d) \neq \emptyset \iff z_{LP}(x) = \begin{cases} \min_k G_k^T(b - Ax) & \textit{falls } H^T(b - Ax) \geq 0 \\ -\infty & \textit{sonst} \end{cases}$

Bemerkung: Für $D(d) \neq \emptyset$ gilt also:

$$z = \max\{c^t x + \min_{k \in K} G_k^T(b - Ax) | H^T(b - Ax) \geq 0, x \in X\} \tag{10.13}$$

Satz 10.13 *(Bender)*

$$\begin{array}{rllcl} z = \max & \eta & & & \\ \text{unter} & \eta \leq c^T x & + \; G_k^T(b - Ax) & & & \forall k \in K \\ & & H^T(b - Ax) & \geq & 0 & \forall j \in J \\ & & x & \in & X & \\ & & \eta & \in & \boldsymbol{R} & \end{array} \tag{10.14}$$

Beweis.

1. Fall: $\{x \in X | H^T(b - Ax) \geq 0\} = \emptyset$. Daraus folgt $z_{LP}(x) = -\infty$ für alle $x \in X$ und damit $z = -\infty$.

2. Fall: $\{x \in X | H^T(b - Ax) \geq 0\} \neq \emptyset$ und $D(d) = \emptyset$. Es folgt $G = \emptyset$ und daher $z = \infty$.

3. Fall: $\{x \in X | H^T(b - Ax) \geq 0\} \neq \emptyset$ und $D(d) \neq \emptyset$: (10.13) und (10.14) sind äquivalent.

□

Bemerkung: Die Relaxation durch Zeilen(Restriktionen)-Generierung führt auf "weniger" Restriktionen.

Beispiel:

$$\begin{array}{lrcrcrcrcrcrcl} \max & 5x_1 & - & 2x_2 & + & 9x_3 & + & 2y_1 & - & 3y_2 & + & 4y_3 & & \\ \text{unter} & 5x_1 & - & 3x_2 & + & 7x_3 & + & 2y_1 & + & 3y_2 & + & 6y_3 & \leq & -2 \\ & 4x_1 & + & 2x_2 & + & 4x_3 & + & 3y_1 & - & y_2 & + & 3y_3 & \leq & 10 \\ & & & & & & & & & & & x_j & \leq & 5 \quad \text{für } j = 1,2,3 \\ & & & & & & & & & & & x & \in & \boldsymbol{Z}_+^3 \\ & & & & & & & & & & & y & \in & \boldsymbol{R}_+^3 \end{array}$$

Sei $X = \{x \in \boldsymbol{Z}_+^3 | x_j \leq 5 \text{ für } j = 1,2,3\}$.

Abbildung II.51 zeigt das Polyeder $\{u \in \boldsymbol{R}_+^2 | uG \geq h\}$:

$$\begin{array}{rcrcr} 2u_1 & + & 3u_2 & \geq & 2 \\ 3u_1 & - & u_2 & \geq & -3 \\ 6u_1 & + & 3u_2 & \geq & 4 \\ & & u & \in & \boldsymbol{R}^2_+ \end{array}$$

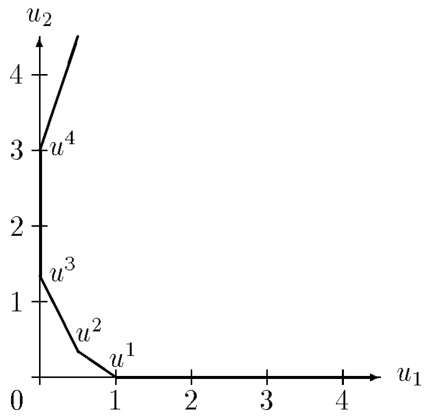


Abbildung II.51: Das Polyeder $\{u \in \boldsymbol{R}^2_+ | uG \geq h\}$

Extremalpunkte: $u^1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$, $u^2 = \begin{pmatrix} \frac{1}{2} \\ \frac{1}{3} \end{pmatrix}$, $u^3 = \begin{pmatrix} 0 \\ \frac{4}{3} \end{pmatrix}$, $u^4 = \begin{pmatrix} 0 \\ 3 \end{pmatrix}$.

Extremalrichtungen: $v^1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$, $v^2 = \begin{pmatrix} 1 \\ 3 \end{pmatrix}$.

Die Umformulierung des gemischt-ganzzahligen linearen Programmes lautet:

max η

unter

$$\begin{array}{lllll} \eta & \leq 5x_1 - 2x_2 + 9x_3 & +(-2 - 5x_1 + 3x_2 - 7x_3) & & \\ \eta & \leq 5x_1 - 2x_2 + 9x_3 & +\frac{1}{2}(-2 - 5x_1 + 3x_2 - 7x_3) & +\frac{1}{3}(10 - 4x_1 - 2x_2 - 4x_3) & \\ \eta & \leq 5x_1 - 2x_2 + 9x_3 & & +\frac{4}{3}(10 - 4x_1 - 2x_2 - 4x_3) & \\ \eta & \leq 5x_1 - 2x_2 + 9x_3 & & +3(10 - 4x_1 - 2x_2 - 4x_3) & \\ & & (-2 - 5x_1 + 3x_2 - 7x_3) & & \geq 0 \\ & & (-2 - 5x_1 + 3x_2 - 7x_3) & +3(10 - 4x_1 - 2x_2 - 4x_3) & \geq 0 \\ & & & x_1, x_2, x_3 & \leq 5 \\ & & & x & \in \boldsymbol{Z}^3_+ \\ & & & \eta & \in \boldsymbol{R}^1 \end{array}$$

Eine Optimallösung ist $x = \begin{pmatrix} 0 \\ 3 \\ 1 \end{pmatrix}$ und $\eta = 3$.

Standortprobleme

$$
(10.15) \qquad \begin{array}{rrrrll}
z = \max & -\sum f_j x_j & + & \sum c_{ij} y_{ij} & & \\
\text{unter} & & & \sum_j y_{ij} & = & 1 \qquad (i \in I) \\
& -x_j & + & y_{ij} & \leq & 0 \qquad (i \in I, j \in N) \\
& & & x & \in & \{0,1\}^n \\
& & & y & \in & \boldsymbol{R}_+^{mn}
\end{array}
$$

Hierbei kann man sich unter f_j die Kosten vorstellen, die bei einer Produktion am Standort j entstehen, und unter c_{ij} den Gewinn, der bei Lieferung vom Produktionsstandort j zum Verkaufsstandort i entsteht.

Das LP(x) hierzu lautet

$$
(10.16) \qquad \begin{array}{rrll}
z_{LP}(x) = \max & \sum c_{ij} y_{ij} & & \\
& \sum_j y_{ij} & = & 1 \\
& y_{ij} & \leq & x_j \quad (i \in I, j \in N) \\
& y & \geq & 0
\end{array}
$$

Die Zerlegung in Teilprobleme zu $i \in I$ liefert:

$$
(10.17) \qquad \begin{array}{rrll}
z_{LP}^i(x) = \max & \sum_j c_{ij} y_{ij} & & \\
& \sum_j y_{ij} & = & 1 \\
& 0 \leq y_{ij} & \leq & x_j \quad (j \in N)
\end{array}
$$

mit $\sum z_{LP}^i(x) = z_{LP}(x)$.
Für $x \neq 0$, was gleichbedeutend mit $\sum_j x_j \geq 1$ ist, sind diese Teilprobleme beschränkt und zulässig.

$$
D^i(c) = \left\{ \begin{pmatrix} u_i \\ v_i \end{pmatrix} \in \boldsymbol{R}^{n+1} | u_i + w_{ij} \geq c_{ij}, w_{ij} \geq 0, j \in N \right\}
$$

hat die Extrempunkte

$$
\begin{bmatrix} c_{ik} \\ (c_{i1} - c_{ik})^+ \\ (c_{in} - c_{ik})^+ \end{bmatrix} \quad (k \in N).
$$

Also gilt

$$
z_{LP}^i(x) = \min_{k \in N} [c_{ik} + \sum_{j \in N} (c_{ij} - c_{ik})^+ x_j]
$$

Nach Satz 10.13 (mit η_i statt η) gilt daher:

$$
\begin{array}{rrcll}
 & z = \max \quad -\sum_{j\in N} f_j x_j + \sum_{i\in I} \eta_i & & & \\
 & \eta_i & \leq & c_{ik} + \sum_{j\in N}(c_{ij} - c_{ik})^+ x_j & \forall i, k \\
(10.18) & \sum x_j & \geq & 1 & \\
 & x & \in & \{0,1\}^n & \\
 & \eta & \in & \boldsymbol{R}^m &
\end{array}
$$

Man erhält also höchstens $m + n + 1$ Restriktionen, ohne Benders Dekomposition hätte man exponentiell viele.

Literaturverzeichnis

[1] Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin. *Network Flows*. Prentice-Hall, Inc., Englewood Cliffs, New Jersey 07632, 1993.

[2] V. Chviátal. *Linear Programming*. W.H.Freeman and Company, New York, 1983.

[3] Thomas H. Cormen, Charles E. Leiserson, and Ronald L. Rivest. *Introduction to Algorithms*. MIT Press, McGraw-Hill, Cambridge, London, 1990.

[4] Shimon Even. *Graph Algorithms*. Computer Science Press, 1979.

[5] Philip E. Gill, Walter Murray, and Margaret H. Wright. *Practical Optimization*. Academic Press, Inc., London, 1981.

[6] M. Gondran and M. Minoux. *Graphs and Algorithms*. John Wiley & Sons, Chichester, 1984.

[7] Martin Grötschel, László Lovász, and Alexander Schrijver. *Geometric Algorithms and Combinatorial Optimization*. Springer, Berlin, 1988.

[8] D. Jungnickel. *Graphen, Netzwerke und Algorithmen*. BI Wissenschaftsverlag, Mannheim, 2 edition, 1990. 2. überarbeitete und erweiterte Auflage.

[9] Eugene L. Lawler. *Combinatorial Optimization: Networks and Matroids*. Holt, Rinehart and Winston, New York, 1976.

[10] J. Leeuven, v. *Algorithms and Complexity*. Elsevier, Amsterdam, 1990.

[11] Th. Lengauer. *Combinatorial Algorithms for Integrated Circuit Layout*. John Wiley & Sons, Chichester, 1990.

[12] Kurt Mehlhorn. *Data Structures and Algorithms: 3. Multidimensional Searching and Computational Geometry*. Springer, Berlin, 1984.

[13] M. Minoux. *Mathematical Programming, Theory and Algorithms*. John Wiley & Sons, Chichester, 1986.

[14] B. M. E. Moret and H. D. Shapiro. *Algorithms from P to NP*. Benjamin Cummings, Redwood City, 1991.

[15] Katta G. Murty. *Network Programming.* Prentice-Hall, Inc., Englewood Cliffs, New Jersey 07632, 1992.

[16] K.G. Murty. *Linear Programming.* John Wiley & Sons, New York/Chichester, 1983.

[17] K.G. Murty. *Linear Complementarity, Linear and Nonlinear Programming.* Heldermann, Berlin, 1988.

[18] George L. Nemhauser and Laurence A. Wolsey. *Integer and Combinatorial Optimization.* John Wiley & Sons, Chichester, 1988. Wiley Interscience Series in Discrete Mathematics and Optimization.

[19] G.L. Nemhauser, A.H.G. Rinnooy Kan, and M.J. Todd. *Optimization.* North-Holland, Amsterdam, 1989.

[20] Christos H. Papadimitriou and Kenneth Steiglitz. *Combinatorial Optimization: Algorithms and Complexity.* Prentice-Hall, Inc., Englewood Cliffs, New Jersey 07632, 1982.

[21] R. G. Parker and L. R. Rardin. *Discrete Optimization.* Academic Press, London, 1988.

[22] H.M. Salkin and K. Mathur. *Foundations for Integer Programming.* North-Holland, New York, 1989.

[23] Alexander Schrijver. *Theory of Linear and Integer Programming.* John Wiley & Sons, Chichester, 1986.

[24] J. Stoer. *Einfuehrung in die Numerische Mathematik I.* Springer, Berlin, 1989.