Algorithm 5: Network Simplex Algorithm

Input: A MIN-COST-FLOW instance (G, u, b, c);
A strongly feasible spanning tree structure (r, T, L, U).

Output: A minimum-cost flow f.

1 Compute b-flow f and potential π associated to (r, T, L, U);
2 $e_0 :=$ an edge with $e_0 \in L$ and $c_\pi(e_0) < 0$ or with $e_0 \in U$ and $c_\pi(e_0) > 0$;
3 **if** (No such edge exists) **then return** f
4 $C :=$ the fund. circuit of e_0 (if $e_0 \in L$) or of $\overleftarrow{e_0}$ (if $e_0 \in U$) and let $\rho = c_\pi(e_0)$;
5 $\gamma := \min_{e' \in E(C)} u_f(e')$.
6 $e' :=$ last edge on C with $u_f(e') = \gamma$ when C is traversed starting at the peak;
7 Let e_1 be the corresponding edge in G, i.e. $e' = e_1$ or $e' = \overleftarrow{e_1}$;
8 Remove e_0 from L or U;
9 Set $T = (T \cup \{e_0\}) \setminus \{e_1\}$;
10 **if** $e' = e_1$ **then** Set $U = U \cup \{e_1\}$;
11 **else** Set $L = L \cup \{e_1\}$;
12 Augment f along γ by C;
13 Let X be the connected component of $(V(G), T \setminus \{e_0\})$ that contains r;
14 **if** $e_0 \in \delta^+(X)$ **then** Set $\pi(v) = \pi(v) + \rho$ for $v \in V(G) \setminus X$;
15 **if** $e_0 \in \delta^-(X)$ **then** Set $\pi(v) = \pi(v) - \rho$ for $v \in V(G) \setminus X$;
16 **go to** line 2;

Theorem

Der Netzwerk-Simplex-Algorithmus terminiert nach endliche vielen Schritten und berechnet eine Optimallösung.

Beweis (Fortsetzung):

z.z.: Der Algorithmus terminiert.

Wir zeigen dazu: Es wird nie dieselbe Baumstruktur zweimal betrachtet.

Die Kosten des Flussnetzes ändern sich um $\gamma \cdot |g|$. $\Rightarrow$ Falls $\gamma > 0$, sind wir fertig.

Also: wir nehmen $\gamma = 0$ an.

Falls $e_0 \neq e_1$, dann gilt $e_0 \in L \cap \delta^-(x)$

oder $e_0 \in U \cap \delta^+(x)$ $\Rightarrow \sum_{v \in V(G)} \pi(v)$ wird

größer und (solange wir nicht um einen

positiven Wert augmentieren) nie kleiner.

Also sei $e_0 = e_1$. Dann gilt $x = v(G)$ und

$\sum_{v \in V(G)} \pi(v)$ bleibt unverändert.

Aber: $|\{e \in L : c_\pi(e) < 0\}| + |\{e \in U : c_\pi(e) > 0\}|$

wird um 1 kleiner.

$\Rightarrow$ Wir betrachten keine Baumstrukturen zweimal. $\square$

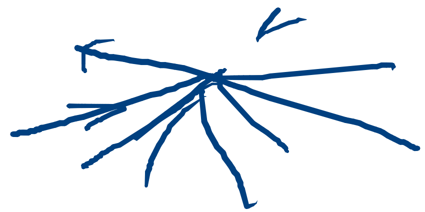
Bestimmung einer stark zulässigen Startlösung:

Verbinde r mit „hinreichend teuren" Kanten mit jeden Knoten.

Für jede Senke $v \in V(G) \setminus \{r\}$ füge eine Kante (r,v) mit $u((r,v)) = -b(v)$ ein. Für alle andere Knoten füge eine Kante (v,r) mit $u((v,r)) = b(v)+1$ ein.

Setze nun L auf die Menge aller Kanten und $U := \emptyset$.

$\Rightarrow$ stark zulässige



Spannbaum-Strukturen.

Polynomielle Algorithmen

Größen von Lösungen

Definiton von Darstellungsgrößen

- $n \in \mathbb{Z}$: $\text{size}(n) := 1 + \lceil \log(|n| + 1) \rceil$,
- $r = \frac{p}{q}$ mit $p, q \in \mathbb{Z}$ teilerfremd: $\text{size}(r) := \text{size}(p) + \text{size}(q)$,
- $x = (x_1, \ldots, x_n) \in \mathbb{Q}^n$: $\text{size}(x) := n + \sum_{i=1}^{n} \text{size}(x_i)$,
- $A = (a_{ij})_{\substack{i=1,\ldots,m \\ j=1,\ldots,n}} \in \mathbb{Q}^{m \times n}$: $\text{size}(A) := nm + \sum_{i=1}^{m} \sum_{j=1}^{n} \text{size}(a_{ij})$.

Annahme: Wir nehmen an, dass Brüche, die irgendwann auftreten, stets sofort mit dem EUKLIDISCHEN ALGORITHMUS gekürzt werden.

Größen von Lösungen

Satz:

Für $r_1, \ldots, r_n \in \mathbb{Q}$ gilt

(a) $\operatorname{size}\left(\prod_{i=1}^{n} r_i\right) \leq \sum_{i=1}^{n} \operatorname{size}(r_i)$

(b) $\operatorname{size}\left(\sum_{i=1}^{n} r_i\right) \leq 2 \sum_{i=1}^{n} \operatorname{size}(r_i)$

<u>Beweis:</u> Beide Aussage sind trivial, falls die $r_1, \ldots, r_n$ natürliche Zahlen sind.
Wir nehmen an, dass $r_i = \frac{p_i}{q_i}$ (p_i, q_i teilerfremd) für $i \in \{1, \ldots, n\}$ gilt.

(a) $\operatorname{size}\left(\prod_{i=1}^{n} r_i\right) \leq \operatorname{size}\left(\prod_{i=1}^{n} p_i\right) + \operatorname{size}\left(\prod_{i=1}^{n} q_i\right)$

$$\leq \sum_{i=1}^{n} \operatorname{size}(p_i) + \sum_{i=1}^{n} \operatorname{size}(q_i)$$

$$= \sum_{i=1}^{n} \operatorname{size}(r_i)$$

(b) $\operatorname{size}\left(\prod_{i=1}^{n} q_i\right) \leq \sum_{i=1}^{n} \operatorname{size}(q_i) \leq \sum_{i=1}^{n} \operatorname{size}(r_i)$

Und: $\operatorname{size}\left(\sum_{i=1}^{n} p_i \prod_{j\in\{1,\dots,n\}\setminus\{i\}} q_j\right)$

$$\leq \operatorname{size}\left(\sum_{i=1}^{n} (p_i / \prod_{j=1}^{n} q_j)\right) \leq \sum_{i=1}^{n} \operatorname{size}(r_i)$$

wegen $\sum_{i=1}^{n} r_i = \frac{1}{\prod_{i=1}^{n} q_i} \sum_{i=1}^{n} p_i \cdot \prod_{j\in\{1,\dots,n\}\setminus\{i\}} q_j$

folgt die Behauptung. $\square$

Größen von Lösungen

Satz:

Für $x, y \in \mathbb{Q}^n$ gilt

(a) $\text{size}(x + y) \leq 2(\text{size}(x) + \text{size}(y))$

(b) $\text{size}(x^t y) \leq 2(\text{size}(x) + \text{size}(y))$

Beweis: (a) $\text{size}(x+y) = n + \sum_{i=1}^{n} \text{size}(x_i + y_i)$

$\leq n + 2\sum_{i=1}^{n} \text{size}(x_i) + 2\sum_{i=1}^{n} \text{size}(y_i)$

$= 2(\text{size}(x) + \text{size}(y)) - 3n$

(6) $$\begin{aligned}\operatorname{size}(x^t y) &= \operatorname{size}\Big(\sum_{i=1}^{n} x_i y_i\Big) \\ &\le 2\sum_{i=1}^{n} \operatorname{size}(x_i y_i) \\ &\le 2\Big(\sum_{i=1}^{n} \operatorname{size}(x_i) + \sum_{i=1}^{n} \operatorname{size}(y_i)\Big) \\ &= 2\big(\operatorname{size}(x) + \operatorname{size}(y)\big) - 4n \quad \square\end{aligned}$$

Größen von Lösungen

Satz:

Für jede Matrix $A \in \mathbb{Q}^{n \times n}$ gilt $\text{size}(\det(A)) \leq 2\text{size}(A)$.

<u>Beweis:</u> Übung.

Größen von Lösungen

Satz:

Sei $\max\{c^t x \mid Ax \leq b\}$ ein zulässiges und beschränktes lineares Programm mit $A \in \mathbb{Q}^{m \times n}$ und $b \in \mathbb{Q}^m$. Dann gibt es eine optimale (rationale) Lösung x mit $\text{size}(x) \leq 4n(\text{size}(A) + \text{size}(b))$. Wenn $b = e_i$ oder $b = -e_i$ für einen Einheitsvektor e_i gilt, dann gibt es ein reguläre Teilmatrix A' von A und eine Optimallösung x mit $\text{size}(x) \leq 4n\text{size}(A')$.

<u>Beweis:</u> Das Optimum wird in einer minimalen Fläche F von $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ angenommen.

Wir können F schreiben als

$F = \{x \in \mathbb{R}^n : \hat{A}x = \hat{b}\}$ für ein Teilsystem

$\hat{A}x \leq \hat{b}$ von $Ax \leq b$

Können annehmen: Die Zeilen von $\hat{A}$ sind linear unabhängig.

Wähle $B \subseteq \{1, \dots, n\}$, sodass $\hat{A}_B$ eine quadratische reguläre Matrix ist.

$\Longrightarrow x$ mit $x_B = \tilde{A}_B^{-1}\tilde{b}$ und x_N

(mit $N = \{1, \dots, n\} \setminus B$) ist eine

optimale LP-Lösung

Nach der Cramerschen Regel können

die Einträge von x_B als

$$x_j = \frac{\det(\hat{A}_j)}{\det(\tilde{A}_B)}$$ geschrieben werden, wobei:

$\hat{A}_j$ aus $\tilde{A}_B$ entsteht, indem die j-te

Spalte durch $\tilde{b}$ ersetzt wird.

$$\Rightarrow \operatorname{size}(x) \leq 4n + 2n\left(\operatorname{size}(\tilde{A}_j) + \operatorname{size}(\tilde{A}_B)\right)$$

$$\leq 4n\left(\operatorname{size}(\tilde{A}_B) + \operatorname{size}(\tilde{b})\right)$$

Falls $b \in \{e_i, -e_i\}$, dann ist

$|\det(\tilde{A}_j)|$ der Absolutbetrag einer

Determinante einer Teilmatrix von $\tilde{A}_B$

$$\begin{array}{cc} & j \\ \left[\begin{array}{c} \tilde{A}_B \end{array}\right. & \left.\begin{array}{c} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{array}\right] \end{array} \quad \longleftrightarrow$$

Größen von Lösungen

Korollar:

Sei $\max\{c^t x \mid Ax \leq b\}$ ein zulässiges und beschränktes lineares Programm mit $A \in \mathbb{Q}^{m \times n}$ und $b \in \mathbb{Q}^m$. Dann gibt es eine optimale (rationale) Lösung x, sodass für jeden Nicht-Null-Eintrag x_j von x gilt: $|x_j| \geq 2^{-4n(\text{size}(A)+\text{size}(b))}$.

Beweis: Es gibt eine Optimallösung x, sodass für jeden Eintrag x_j von x gilt: $\text{size}(x_j) \leq 4n\left(\text{size}(A) + \text{size}(b)\right)$

Da jede positive Zahl, die kleiner als $2^{-4n(size(A)+size(b))}$ ist eine size hat, die größer als $-4n(size(A)+size(b))$ ist, folgt die Behauptung. $\square$

Gauss-Elimination

Wir wollen ein Gleichungssystem $Ax = b$ lösen.

Transformiere A dazu in eine obere rechte Dreicksmatrix mit folgenden erlaubten Schritten:

- Addiere ein Vielfaches einer Zeile zu einer anderen.
- Vertausche zwi Spalten.
- Vertausche zwei Zeilen.

Wir wissen aus Alma I: Es reichen dabei $O(mn(\text{rank}(A) + 1))$ elementare Operationen aus.

Ziel: Zeige polynomielle Laufzeit.

Zu zeigen: Alle auftretenden Zahlen können mit polynomiell vielen Bits geschrieben werden.

Sei: $\tilde{A} = \begin{pmatrix} B & C \\ 0 & D \end{pmatrix}$ eine Matrix, die während der Gauss-Elimination betrachtet wird. Also ist B eine rechte obere $k \times k$-Dreiecksmatrix

Wir können annehmen, dass wir weder Zeilen noch Spalten vertauschen müssen.

Für jeden Eintrag d_{ij} von D gilt

$$\det\left(\tilde{A}^{1\ldots k,\, k+i}_{1\ldots k,\, k+j}\right) = d_{ij} \cdot \det\left(\tilde{A}^{1\ldots k}_{1\ldots k}\right)$$

wobei $\left(M^{i_1\ldots i_t}_{j_1\ldots j_t}\right)$ die Untermatrix einer Matrix M mit Zeilenindizes $i_1 \ldots i_t$ und Spaltenindizes $j_1 \ldots j_t$ sei:

$$\tilde{A}^{1\ldots k,\, k+i}_{1\ldots k,\, k+j} = \begin{bmatrix} B & \vdots \\ 0 \cdots 0 & d_{ij} \end{bmatrix}$$

Das Addieren eines Vielfachen einer Zeile zu einer andern ändert nichts an Determinanten.

$\Rightarrow$ d_{ij} (und damit jede Zahl, die bei der Gauss-Elimination auftritt) kann als Quotient von Determinanten von Untermatrizen von A geschrieben werden.

$\Rightarrow$ Alle vorkommenden Zahlen haben polynomielle Größe.

$\Rightarrow$ Gauss-Elimination ist polynomiell.