

Kombinatorische Optimierung

Approximation und Randomisierung

private Mitschrift – *kein* offizielles Script – www.kuertz.net/uni

Id: KombinatorischeOptimierung2.tex,v 1.21 2005/07/01 11:16:48 mtu Exp

Disclaimer: Dies ist kein offizielles Script, sondern nur eine private Mitschrift. Ich kann daher keine Gewähr für die Richtigkeit übernehmen. Vor allem weicht die Nummerierung der Kapitel und Sätze zum Teil von der in der Vorlesung verwendeten ab. Falls jemand einen Fehler entdeckt, so möge er/sie mir bitte eine eMail schicken - vielen Dank!

Max Tuengerthal (max@17q.de)

Inhaltsverzeichnis

11 Lineare Programmierung	1
11.1 Farkas Lemma	1
11.2 Der Dualitätssatz der linearen Programmierung	6
12 Lineare Programmierung - Algorithmen	11
13 Travelling-Salesman-Problem	16
13.1 Algorithmus von Christofides	17
13.2 Insert-Heuristiken	18
13.3 Approximationsschema von Arora	19
14 Steinerbäume	24
15 Mehrgüterflüsse	30
15.1 Fraktionale und ganzzahlige Mehrgüterflüsse	30
15.2 Ein 1/2-Faktor Approximationsalgorithmus für Bäume	31
16 Packen und Überdecken in Hypergraphen	36
16.1 Chernoff-Hoeffding-Ungleichungen	36
16.2 Matching in Hypergraphen	38
17 Überdeckungen in Hypergraphen	43
18 Max-Cut Problem	48
19 Dichte Subgraphen	53
20 Probabilistisch verifizierbare Beweise und Nicht-Approx.	60
20.1 Die Klasse PCP(r,q)	60
21 Färben 3-färbbarer Graphen mit semidefiniter Relaxierung	63

11 Lineare Programmierung

Problem: Maximiere

$$c^T x$$

unter der Nebenbedingung

$$Ax \leq b$$

$(x \in \mathbb{R}^n, A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m)$.

Fragen:

1. Struktur der Lösung
2. Effiziente Algorithmen (historisch ausserordentlich wichtig)
- 2'. Test, ob $P = \{x \in \mathbb{R}^n \mid Ax \leq b\} \neq \emptyset$
3. Was passiert bei ganzzahligen Lösungen, d.h. $x \in \mathbb{R}^n$?
(NP-schwer)

11.1 Farkas Lemma

Eine Menge $K \subseteq \mathbb{R}^n$ heißt *konvex*, wenn für alle $x, y \in K$, $\lambda \in [0, 1]$ gilt:
 $\lambda x + (1 - \lambda)y \in K$.

Ein Vektor $x \in \mathbb{R}^n$ heißt *konvexe Kombination* von $y_1, \dots, y_n \in \mathbb{R}^n$, falls $\lambda_1, \dots, \lambda_n \in [0, 1]$ ex. mit $\sum_{i=1}^n \lambda_i = 1$ und $x = \sum_{i=1}^n \lambda_i y_i$.

Für eine Menge $X \subseteq \mathbb{R}^n$ sei $\text{co}(X)$ die *konvexe Hülle* von X . (= alle Vektoren, die Konvexkombination von Vektoren aus X sind).

Insbesondere: $\{x \in \mathbb{R}^n \mid Ax \leq b\}$, $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$ ist eine konvexe Menge.

Mengen dieser Form heissen *Polyeder*, und *Polytop*, wenn sie normbeschränkt sind.

Halbraum: $\{x \in \mathbb{R}^n \mid (Ax)_i \leq b_i\}$.

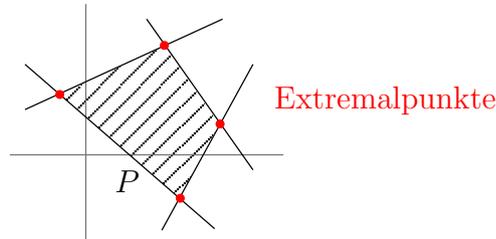
Hyperebene: $H = \{x \in \mathbb{R}^n \mid x = c, c \in \mathbb{R}\}$.

Definition 11.1. Sei $K \subseteq \mathbb{R}^n$ konvex, $x \in K$ heisst *Extremalpunkt* oder *Ecke* von K , falls aus der Darstellung $x = \lambda y + (1 - \lambda)z$, $\lambda \in (0, 1)$, $y, z \in K$, stets folgt: $x = y = z$.

Mit $\text{ex}(K)$ bezeichnen wir die Menge der Extremalpunkte von K .

Beispiel. 1. Die Menge \mathbb{R}^n besitzt keine Extrempunkte.

2. Für Polyeder gilt $P = \overline{\text{co}(\text{ex}(P))}^{\|\cdot\|}$ (Satz v. Krein-Milman).



Proposition 11.2. Sei $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m \setminus \{0\}$. Genau eine der beiden Möglichkeiten trifft zu:

- (i) $Ax = b$, $x \in \mathbb{R}^n$ lösbar.
- (ii) $A^T y = 0$, $b^T y = -1$, $y \in \mathbb{R}^m$ ist lösbar.

Beweis. Annahme: (i) und (ii) sind gleichzeitig erfüllt mit Lösungen x, y .
Dann:

$$0 = x^T A^T y = (Ax)^T y \stackrel{(i)}{=} b^T y \stackrel{(ii)}{=} -1 \quad \text{Widerspruch!}$$

Wir zeigen: (ii) gilt, falls (i) nicht gilt.

Sei $A' = (A \ b)$, $c = (0, \dots, 0, -1)^T \in \mathbb{R}^{m+1}$. $U = \text{Lin}(a^1, \dots, a^n)$, $U' = \text{Lin}(a'_1, \dots, a'_m, a'_{m+1})$, a^i ist i -ter Spaltenvektor von A , a'_i ist i -ter Zeilenvektor von A' . (Lin – Vektorraum-Erzeugnis)

Falls (i) nicht gilt, d.h. $Ax = b$ ist nicht lösbar, so gilt: $\text{rang}(A') = \text{rang}(A) + 1$ und $c \in U'$ (gdw. (ii) gilt). □

Satz 11.3 (Farkas Lemma 1). Genau eine der beiden Möglichkeiten trifft zu:

- (i) $Ax = b$, $x \geq 0$, $x \in \mathbb{R}^n$ ist lösbar.
- (ii) $A^T y \geq 0$, $b^T y < 0$, $y \in \mathbb{R}^m$ ist lösbar.

Beweis. (i) und (ii) sind Alternativen (wie in 11.2).

Wir zeigen: $\neg(i) \Rightarrow (ii)$.

Dann ist $b \neq 0$.

- a) $Ax = b$ ist nicht lösbar. Dann folgt die Beh. mit Prop. 11.2.
- b) $Ax = b$ ist lösbar, aber $x \not\geq 0$. Folgere (ii) per Induktion:

I.A.: $n = 1$. Sei $x \in \mathbb{R}$ Lösung von $Ax = b$, $x \not\geq 0$. D.h. $x < 0$. Mit $y := -b$ folgt: $A^T y = -\frac{1}{x} \underbrace{b^T b}_{\geq 0} \geq 0$ (wegen $Ax = b$), $b^T y = -y^T y = -y^2 < 0$.

I.S.: Beh. gelte f.a. $k \leq n - 1$, $n \geq 2$. Sei $A' = [a^1, \dots, a^{n-1}]$. Dann ist $A'x = b$, $x \geq 0$ in \mathbb{R}^{n-1} nicht lösbar gdw.

a) $A'x = b$ ist nicht lösbar oder

b) $A'x = b$ ist lösbar, aber $x \not\geq 0$.

Aus (a) folgt mittels Prop. 11.2 aus der Induktionsannahme: $A^T y \geq 0$, $b^T y < 0$ ist lösbar. Sei $v \in \mathbb{R}^m$ eine Lösung, d.h. $A^T v \geq 0$, $b^T v < 0$.

z.z.: $A^T v = A^T v$. Falls $(a^n)^T v \geq 0$ gilt, sind wir fertig.

Annahme: $(a^n)^T v < 0$.

Wir stellen ein System $\bar{A}x = \bar{b}$, $x \geq 0$, $x \in \mathbb{R}^{n-1}$ auf und diskutieren die Alternativen: das System ist lösbar, bzw. nicht lösbar.

$\bar{a}^i := \alpha_i a^n - \alpha_n a^i$ für $i = 1, \dots, n - 1$

$\bar{b} := \beta a^n - \alpha_n b$.

$\alpha_i := (a^i)^T v$, $\beta := b^T v$.

1. Fall: Obiges System ist lösbar mit Lösung x .

$$\beta a^n - \alpha_n b = \bar{b} = \bar{A}x = \sum_{i=1}^n \bar{a}^i x_i = \sum_{i=1}^n (\alpha_i a^n - \alpha_n a^i) x_i$$

$$\Rightarrow b = \left(\frac{\beta}{\alpha_n} - \frac{1}{\alpha_n} \sum_{i=1}^{n-1} x_i \alpha_i \right) a^n + \sum_{i=1}^{n-1} x_i a^i = Ax^*,$$

mit $x_i^* = x_i$ für alle $i \leq n - 1$ und $x_n^* = \left(\frac{\beta}{\alpha_n} - \frac{1}{\alpha_n} \cdot \sum_{i=1}^{n-1} \alpha_i x_i \right)$.
z.B.: $x_n^* \geq 0$.

Für die ersten $(n - 1)$ Komponenten ist dies klar, für die n -te folgt dies aus der Definition von x_n^* . $\nexists Ax = b$ nicht lösbar.

2. Fall: Das obige System ist nicht lösbar.

Konstruiere Lösung y von (ii) direkt mittels der Induktionsannahme. □

Satz 11.4 (Farkas Lemma 2). Genau eine der Möglichkeiten trifft zu:

(i) $Ax \leq b$, $x \geq 0$, $x \in \mathbb{R}^n$ ist lösbar.

(ii) $A^T y \geq 0$, $y \geq 0$, $b^T y < 0$, $y \in \mathbb{R}^m$ ist lösbar.

Beweis. (i), (ii) sind nicht gleichzeitig lösbar (wie Farkas Lemma 1 (11.3)).

Angenommen (i) nicht erfüllt.

$\Rightarrow Ax + z = b, x \geq 0, z \geq 0$ ist nicht lösbar. Sei E_m die Einheitsmatrix im $\mathbb{R}^{m \times m}$, und $B = (A|E_m)$.

$Bw = b, w \geq 0$ ist nicht lösbar *gdw.* obiges System nicht lösbar.

Mit Farkas Lemma 1 folgt, dass ein $y \in \mathbb{R}^m$ existiert, so dass $A^T y \geq 0$, $E_m y = y \geq 0$ und $b^T y < 0$. \square

Satz 11.5 (Farkas Lemma 3). *Genau eine der Möglichkeiten trifft zu:*

(i) $Ax \leq b, x \in \mathbb{R}^n$ ist lösbar.

(ii) $A^T y = 0, y \geq 0, b^T y < 0, y \in \mathbb{R}^m$ ist lösbar.

Beweis. (i) und (ii) sind Alternativen (wie Farkas Lemma 1 (11.3)).

Definiere

$$A' := (E_m | A | -A)$$

Beh.: (*)

$$A'x' = b, x' \geq 0, x' \in \mathbb{R}^{m+2n} \text{ lösbar}$$

\Leftrightarrow

$$Ax \leq b, x \in \mathbb{R}^n \text{ lösbar}$$

Bew.: " \Rightarrow ": Sei $x' = (x'^1, x'^2, x'^3) \geq 0$ mit $x'^1 \in \mathbb{R}^m, x'^2, x'^3 \in \mathbb{R}^n$.

$$x := x'^2 - x'^3$$

Dann gilt

$$Ax = Ax'^2 - Ax'^3 = b - x'^1 \leq b$$

" \Leftarrow ": Sei x so, dass $Ax \leq b$ ist.

$$x_i'^2 := \begin{cases} x_i & x_i > 0 \\ 0 & \text{sonst} \end{cases} \in \mathbb{R}^n$$

$$x_i'^3 := \begin{cases} -x_i & x_i < 0 \\ 0 & \text{sonst} \end{cases} \in \mathbb{R}^n$$

$$x'^1 := b - Ax'^2 + Ax'^3 \in \mathbb{R}^m$$

$$x' := (x'^1 | x'^2 | x'^3)$$

Also $x' \geq 0$ und

$$A'x' = E_m x'^1 + Ax'^2 + Ax'^3 = b - Ax'^2 + Ax'^3 + Ax'^2 - Ax'^3 = b$$

Sei (i) nicht lösbar. Dann ist nach (*)

$$A'x' = b, x' \geq 0$$

nicht lösbar. Nach Farkas Lemma 1 ist

$$A^T y \geq 0, b^T y < 0 \text{ lösbar}$$

$$0 \leq A^T y = \begin{pmatrix} E_m \\ A^T \\ -A^T \end{pmatrix} y = \begin{pmatrix} y \\ A^T y \\ -A^T y \end{pmatrix}$$

Also $y \geq 0$ und $A^T y = 0$. □

Satz 11.6 (Farkas Lemma 4). Sei $Ax \leq b$, $x \in \mathbb{R}^n$ lösbar. Sei $\delta \in \mathbb{R}$. Jede Lösung x von $Ax \leq b$ erfüllt $c^T x \leq \delta$ gdw. ein $y \in \mathbb{R}^m$ existieren mit $y^T A = c^T$ und $y^T b \leq \delta$, $y \geq 0$.

Beweis. “ \Leftarrow ”: Sei $x \in \mathbb{R}^n$ mit $Ax \leq b$. Sei $y \in \mathbb{R}^m$ mit $y^T A = c^T$ und $y^T b \leq \delta$, $y \geq 0$.
 $\Rightarrow c^T x = y^T Ax \leq y^T b \leq \delta$.

“ \Rightarrow ”: Annahme: $y^T A = c^T$, $y^T b \leq \delta$, $y \geq 0$ ist nicht lösbar.

Dies gilt gdw. $(y^T \quad \lambda) \begin{pmatrix} A & b \\ 0 & 1 \end{pmatrix} = (c^T \quad \delta)$, $(y^T \quad \lambda) \geq (0 \quad 0)$, $\lambda \geq 0$, nicht lösbar (gdw. $x^T A^T = c^T$, $x' \geq 0$).

Farkas La 1 [11.3] \Rightarrow Es ex. $\begin{pmatrix} z \\ N \end{pmatrix}$, $z \in \mathbb{R}^m$, $N \in \mathbb{R}$:

$$\begin{pmatrix} A & b \\ 0 & 1 \end{pmatrix} \begin{pmatrix} z \\ N \end{pmatrix} \geq \begin{pmatrix} 0 \\ 0 \end{pmatrix}, (c^T \quad \delta) \begin{pmatrix} z \\ N \end{pmatrix} < 0.$$

Insbesondere ist $N \geq 0$.

1. Fall: $N = 0$. Dann: $Az \geq 0$, $c^T z < 0$.

Sei x^* eine Lösung von $Ax \leq b$. Sei $\gamma > 0$.

Setze $x^* - \gamma z =: x'$.

$$Ax' = A(x^* - \gamma z) = Ax^* - \gamma Az \leq b - \underbrace{\gamma(Az)}_{\geq 0} \leq b.$$

$c^T x' > \delta$, denn:

$$c^T x' = c^T(x^* - \gamma z) = c^T x^* - \underbrace{\gamma c^T z}_{<0}$$

Widerspruch zur Voraussetzung!

2. Fall: $N > 0$. Somit gilt:

$$Az + Nb \geq 0, \quad c^T z + N\delta < 0 \quad | : (-N)$$

$$A\left(-\frac{z}{N}\right) \leq b, \quad c^T\left(-\frac{z}{N}\right) > \delta \quad \text{Widerspruch!}$$

□

11.2 Der Dualitätssatz der linearen Programmierung

Sei $A \in \mathbb{R}^{m \times n}$ mit Zeilenvektoren a_1, \dots, a_m und Spaltenvektoren a^1, \dots, a^n , sowie $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$. Sei (I, I^c) eine Partition von $\{1, \dots, m\}$, (J, J^c) eine Partition von $\{1, \dots, n\}$. Wir nennen die Variablen x_j für $j \in J^c$ *unbeschränkte Variablen*.

Definition 11.7 (Lineares Programm). 1. Allgemeinform:

$$\begin{array}{ll} \max c^T x & \min c^T x \\ a_i^T x = b_i, i \in I & a_i^T x = b_i, i \in I \\ a_i^T x \leq b_i, i \in I^c & a_i^T x \geq b_i, i \in I^c \\ x_j \geq 0, j \in J & x_j \geq 0, j \in J \\ x_j \in \mathbb{R}, j \in J^c & x_j \in \mathbb{R}, j \in J^c \end{array}$$

2. Standardform

$$\begin{array}{ll} \max c^T x & \min c^T x \\ Ax \leq b & Ax \geq b \\ x \geq 0 & x \geq 0 \end{array}$$

3. Kanonische Form

$$\begin{array}{ll} \max c^T x & \min c^T x \\ Ax = b & Ax = b \\ x \geq 0 & x \geq 0 \end{array}$$

(Überall: $x \in \mathbb{R}^n$)

Proposition 11.8. Die drei Formen des LP (linearen Programmes) sind äquivalent.

Beispiel. Sei $G = (V, E)$ ein Graph. Sei $A = (a_{i,j})_{1 \leq i \leq m, 1 \leq j \leq n}$ die *Kanten/Knoten Inzidenzmatrix* von G . (mit $a_{i,j} = 1$ falls Knoten $j \in$ Kante i , und 0 sonst).

Maximum Matching Problem: Finde $M \subseteq E$ mit M Matching und $|M|$ maximal.

Formulierung als ganzzahliges LP:

$$\begin{aligned} \max \sum_{i=1}^m x_i \\ \sum_{i=1}^m a_{i,j} x_i \leq 1 \quad \forall j \in \{1, \dots, n\} & \quad (\Leftrightarrow A^T x \leq 1) \\ x_i \in \{0, 1\} \quad \forall i \in \{1, \dots, m\} & \quad (\Leftrightarrow x \in \{0, 1\}^m) \end{aligned}$$

MaxMatching $\in P$

LP-Relaxierung: $x \in [0, 1]^m \rightarrow$ fraktionales Matching.

Definition 11.9 (Duales Problem).

Primales Problem (P) $\max c^T x$ $a_i^T x = b_i, i \in I$ $a_i^T x \leq b_i, i \in I^c$ $x_j \geq 0, j \in J$ $x_j \in \mathbb{R}, j \in J^c$ $x \in \mathbb{R}^n$	Duales Problem (P^*) $\min y^T b$ $y_i \in \mathbb{R}, i \in I$ $y_i \geq 0, i \in I^c$ $y^T a^j \geq c_j, j \in J$ $y^T a^j = c_j, j \in J^c$ $y \in \mathbb{R}^m$
---	---

Satz 11.10.

$$(P^{**}) = (P),$$

d.h. das Dual des Duals ist das primale Problem.

Satz 11.11 (Schwacher Dualitätssatz). Seien $x \in \mathbb{R}^n, y \in \mathbb{R}^m$ zulässig für (P) bzw. (P^*) . Dann:

$$c^T x \leq b^T y.$$

Korollar 11.12. Seien x, y zulässig mit $c^T x = b^T y$, dann sind x und y optimal für (P) bzw. (P^*) .

Beweis. Seien x^*, y^* opt. Lsg., dann

$$c^T x^* \geq c^T x = y^T b \geq y^{*T} b \stackrel{\text{Satz 11.11}}{\geq} c^T x^*.$$

□

Satz 11.13 (Dualitätssatz der lin. Programmierung). Seien Primal/-Dual gegeben durch:

$$\begin{array}{ll}
 (P) & (P^*) \\
 \max c^T x & \min b^T y \\
 Ax \leq b & A^T y \geq c \\
 x \geq 0, x \in \mathbb{R}^n & y \geq 0, x \in \mathbb{R}^m
 \end{array}$$

1. Haben (P) und (P^*) zulässige Lösungen, so haben sie auch optimale Lösungen und es gilt

$$OPT(P) = OPT(P^*)$$

2. Hat (P) oder (P^*) keine zulässige Lsg., so besitzt weder (P) noch (P^*) eine optimale Lösung mit endlichem Wert.

Beweis. 1. Es reicht x, y zulässig zu finden mit $c^T x \geq b^T y$. (Denn dann folgt aus dem schwachen Dualitätssatz die Lösung, und irgendwie auch, dass sie optimal sind)

Dies ist äquivalent zu

$$\begin{pmatrix} A & 0 \\ 0 & -A^T \\ -c^T & b^T \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \leq \begin{pmatrix} b \\ -c \\ 0 \end{pmatrix}, \quad \begin{pmatrix} x \\ y \end{pmatrix} \geq 0 \text{ ist lösbar.}$$

Annahme: Obiges System nicht lösbar. Dann folgt mit Farkas Lemma 2 (11.4), dass $z \in \mathbb{R}^m$, $w \in \mathbb{R}^n$ und $\alpha \in \mathbb{R}$ existieren mit $z, w, \alpha \geq 0$ und $A^T z \geq \alpha c$, $Aw \leq \alpha b$, $b^T z < c^T w$.

Beh.: $\alpha > 0$.

Bew.: Angenommen, $\alpha = 0$. Seien x_0, y_0 zulässige Lsg. (nach Vor.)

$$0 \leq x_0^T A^T z = (Ax_0)^T z \leq b^T z < c^T w \leq (A^T y_0)^T w = y_0^T Aw \leq 0 \quad \text{!}$$

Sei $x := \alpha^{-1}w$, $y := \alpha^{-1}z$. Dann folgt: $Ax \leq b$, $A^T y \geq c$, $x \geq 0$, $y \geq 0$, d.h. x, y zulässig für (P) bzw. (P^*) .

$$c^T w = \alpha(c^T x) \stackrel{\text{schwacher DS}}{\leq} \alpha(b^T y) = b^T z \quad \text{Widerspruch!}$$

2. O.B.d.A. habe (P) keine zulässige Lsg., dann folgt mit Farkas Lemma 2 (11.4), dass $w \in \mathbb{R}^m$ existiert, mit $A^T w \geq 0$, $b^T w < 0$, $w \geq 0$. Falls (P^*) eine zulässige Lösung besitzt, sagen wir y_0 , so ist $y_0 + \lambda w$ auch zulässig für (P^*) mit $\lambda \geq 0$. Es folgt:

$$b^T(y_0 + \lambda w) = b^T y_0 + \lambda \underbrace{(b^T w)}_{<0} \xrightarrow{\lambda \rightarrow \infty} -\infty.$$

□

Satz 11.14 (Komplementären Schlupf). Sei x^*, y^* zulässig für (P) bzw. (P^*) . x^*, y^* optimal gilt gdw.

1. für alle j gilt $x_j^* (y^{*T} a^j - c_j) = 0$ und
2. für alle i gilt $y_i^* (b_i - a_i^T x^*) = 0$.

Beweis. $I = \{1, \dots, k\}$, $J = \{1, \dots, l\}$ o.B.d.A.

Es gilt $x_j^* (y^{*T} a^j - c_j) \geq 0$ und $y_i^* (b_i - a_i^T x^*) \geq 0$ für alle i und j .

Sei $t := \sum_{j=1}^n x_j^* (y^{*T} a^j - c_j) + \sum_{i=1}^m y_i^* (b_i - a_i^T x^*)$, damit $t \geq 0$.

Beh.: $t = 0 \Leftrightarrow x^*, y^*$ optimal.

Bew.: Umformen:

$$\begin{aligned} t &= - \sum_{j=1}^n x_j^* c_j + \sum_{i=1}^m y_i^* b_i + \underbrace{\sum_{j=1}^n x_j^* y^{*T} a^j}_{\langle x^*, A^T y^* \rangle} - \underbrace{\sum_{i=1}^m y_i^* a_i^T x^*}_{\langle y^*, A x^* \rangle} \\ &= -y^{*T} b - c^T x^* \end{aligned}$$

Also: $t = 0 \Rightarrow y^{*T} b = c^T x^* \Rightarrow$ (mit Kor 11.12) x^*, y^* opt.

x^*, y^* optimal $\Rightarrow t = 0$. □

Sei $\mathcal{H} = (V, \varepsilon)$ ein *Hypergraph* (oder Mengensystem), d.h. V endliche Menge, $\varepsilon \subseteq \mathcal{P}(V)$. Elemente aus ε heißen Hyperkanten. Für $|E| = 2$, für alle $E \in \varepsilon$, ist \mathcal{H} ein Graph.

Ein k -Matching in \mathcal{H} ist eine Teilmenge $\varepsilon' \subseteq \varepsilon$, so dass in jedem Knoten aus V höchstens k Hyperkanten aus ε inzidieren.

Problem: Maximiere k -Matching in Hypergraphen.

Finde k -Matching ε' mit $|\varepsilon'|$ maximal.

Komplexität: Im “Graphfall” in P . Im “Hypergraphfall” ist es NP-schwer, schon wenn alle Hyperkanten Kardinalität 3 haben.

$$\begin{array}{ll}
 \text{(ILP)} & \text{(LP)} \\
 \max \sum_{i=1}^m x_i & \max \sum_{i=1}^m x_i \\
 \sum_{i=1}^m a_{i,j} x_i \leq k \quad \forall j \in V & \sum_{i=1}^m a_{i,j} x_i \leq k \quad \forall j \in V \\
 x_i \in \{0, 1\} \quad \forall i \in \{1, \dots, m\} & x_i \in [0, 1] \quad \forall i \in \{1, \dots, m\} \\
 & (|\varepsilon| = m, |V| = n)
 \end{array}$$

1. Mit z. B. Ellipsoidalgorithmus löse (LP) optimal mit Lösung $y = (y_1, \dots, y_m) \in [0, 1]^m$.

2'. Naives Suchen.

$$\text{Setze } x_i = \begin{cases} 1 & y_i \geq \frac{1}{2} \\ 0 & \text{sonst} \end{cases} \quad \text{für alle } i.$$

Sei $\varepsilon' = \{i \in \varepsilon \mid x_i = 1\}$.

Frage: Ist ε' ein k -Matching und wie nah ist $|\varepsilon'|$ an OPT (= Kardinalität eines Maximum k -Matching)?

\rightsquigarrow Im Allgemeinen hat ε' nichts mit einem k -Matching zu tun.

2. Randomisiertes Runden (RR).

$$\text{Setze } x_i = \begin{cases} 1 & \text{mit W'keit } y_i \\ 0 & \text{mit W'keit } 1 - y_i \end{cases} \quad \text{für alle } i \text{ unabhängig.}$$

Analyse des RR:

a)

$$\mathbb{E} \left(\sum_{i=1}^m a_{i,j} x_i \right) = \sum_{i=1}^m a_{i,j} \mathbb{E}(x_i) = \sum_{i=1}^m a_{i,j} y_i \leq k$$

b)

$$\mathbb{E} \left(\sum_{i=1}^m x_i \right) = \sum_{i=1}^m \mathbb{E}(x_i) = \sum_{i=1}^m y_i = \text{OPT}(LP) \geq \text{OPT}$$

Angenommen wir könnten zeigen, dass: Mit W'keit $\geq \frac{1}{2}$ gilt $\sum a_{i,j} x_i \leq k$ für alle j und $\sum x_i \geq \alpha \cdot \text{OPT}$, für $\alpha < 1$ konstant. Dann kann mit Derandomisierung, daraus ein polynomzeit Algorithmus erzeugt werden, der das Problem mit etwas schlechterem α löst.

Aufgabe. Gegeben sei ein Museum mit n Wänden.

Frage: Wieviele Wächter benötigt man höchstens, um jeden Punkt im Museum durch mind. einen Wächter zu beobachten?

Antwort: Man benötigt höchstens $\lceil \frac{n}{3} \rceil$ Wächter.

12 Lineare Programmierung – Algorithmen (Ellipsoid-Algorithmus)

1979 Khachiyan: LP \in P

Der Algorithmus zeigt in polynomierller Zeit, ob ein Polyeder leer ist oder nicht. [\Leftrightarrow LP]

Gegeben ist ein Polyeder $P = \{x \in \mathbb{R}^n \mid Ax \leq b\}$.

Frage: $P \neq \emptyset$?

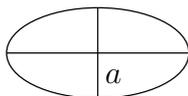
Sei $A \in \mathbb{R}^{n \times n}$ pos. defi., $a \in \mathbb{R}^n$. Wir definieren $\|\cdot\|_A$ durch

$$\|x\|_A := \sqrt{\langle x, A^{-1}x \rangle} \quad (x \in \mathbb{R}^n).$$

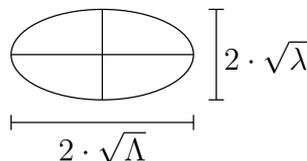
$\|\cdot\|_A$ ist eine Norm in \mathbb{R}^n .

Wenn $A = E_n$ (Einheitsmatrix), so $\|\cdot\|_A = \|\cdot\|_2$. (Ellipsoid-Norm)

Die Menge $E(A, a) = \{x \in \mathbb{R}^n \mid \|x - a\|_A \leq 1\}$ ist ein *Ellipsoid* mit Zentrum a .



Wenn λ kleinster Eigenwert von A und Λ größter Eigenwert von A , dann



$2\sqrt{\lambda}$ – Länge der kleinsten Achse

$2\sqrt{\Lambda}$ – Länge der größten Achse

Wir wissen: Es existiert B positiv definit: $A = B^T B$ (nach Cholesky-Zerlegung).

Man nennt B die Wurzel aus A , Notation $A^{\frac{1}{2}} = B$.

Sei $B(0, 1)$ die Einheitskugel in \mathbb{R}^n . Dann

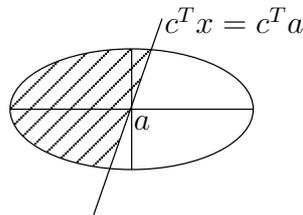
$$E(A, a) = A^{\frac{1}{2}}B(0, 1) + a.$$

Definition. Eine Abb. $T: \mathbb{R}^n \rightarrow \mathbb{R}^n$ heißt *affine Transformation*, falls $T(x) = Qx + q \quad \forall x \in \mathbb{R}^n$, wobei $Q \in \mathbb{R}^{n \times n}$ pos. definit, $q \in \mathbb{R}^n$.

Bemerkung. Ferner gilt: Mit $T: x \mapsto A^{\frac{1}{2}}(x - x_0)$, $x_0 = -A^{-\frac{1}{2}}a$, ist T eine affine Transformation und $E(A, a) = TB(0, 1)$.

Definition. Ein *Halbellipsoid* $E(A, a, c)$ ist der Schnitt des Ellipsoids $E(A, a)$ mit dem Halbraum $H = \{x \in \mathbb{R}^n \mid c^T x \geq c^T a\}$.

Beispiel.

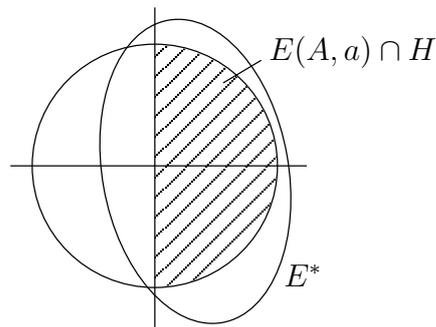


Lemma 12.1 (Shrinking Lemma). Sei $E(A, a) = B(0, 1)$ und $H = \{x \in \mathbb{R}^n \mid x_1 \geq 0\}$. Sei

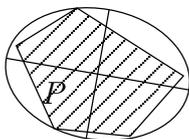
$$E^* = \left\{ x \in \mathbb{R}^n \mid \left(\frac{n+1}{n} \right)^2 \left(x_1 - \frac{1}{n+1} \right)^2 + \frac{n^2-1}{n^2} \sum_{i=2}^n x_i^2 \leq 1 \right\}.$$

E^* ist ein Ellipsoid, das $E(A, a) \cap H$ enthält und es gilt

$$\text{vol}(E^*) = \frac{n}{n+1} \left(\frac{n^2}{n^2-1} \right)^{\frac{n-1}{2}} \underbrace{\text{vol}(E(A, a))}_{=\tau_n} \leq e^{-\frac{1}{2(n+1)}} \cdot \text{vol}(E(A, a)).$$

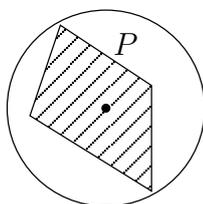


Korollar 12.2. Sei $P = \{x \in \mathbb{R}^n \mid Cx \leq d\}$ ein Polytop, $d \in \mathbb{R}^n$, $C \in \mathbb{R}^{m \times n}$. Das Ellipsoid kleinsten Volumens, das P enthält, hat sein Zentrum in P .

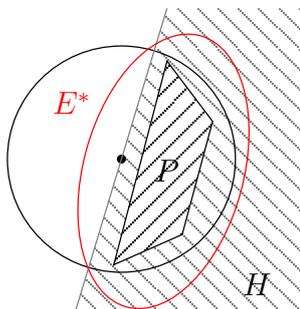


Idee des Ellipsoid-Algorithmus

1. Lege eine kleine Kugel um P , Falls Mittelpunkt der Kugel in P , so sind wir fertig und $P \neq \emptyset$.



2. Andernfalls gibt es eine Ungleichung des Polytops, z.B. $(Cx)_i \leq d_i$ für ein $i \in \{1, \dots, m\}$, die durch a ($=$ Zentrum von $E(A, a)$) verletzt wird. Das bedeutet: Es gibt einen Halbraum H mit $E(A, a) \cap H \supseteq P$.



1. **Fall:** Das Polytop ist nicht leer. Durch das Shrinking Lemma (12.1) erhalten wir immer kleinere Ellipsoide (d.h. mit kleinerem Volumen) und irgendwann muss das Zentrum im Polytop liegen (das können wir leicht testen).
2. **Fall:** Das Polytop ist leer. (Finde ein geometrisches Object \mathcal{O} mit $\mathcal{O} \subseteq P$ und $\text{vol}(\mathcal{O}) \geq K$.) Falls $\text{vol}(E^*) < K \Rightarrow P = \emptyset$.

Ellipsoid-Algorithmus

Eingabe: Polytop $P = \{Cx \leq d\}$. P sei entweder leer oder voll-dimensional (d.h. P enthält $n + 1$ affin-unabhängige Punkte). C, d haben rationale Kom-

ponenten.

Initialisierung:

- a) $k = 0$
- b) $N = 2n((2n+1)\langle C \rangle + n\langle d \rangle - n^3)$
($\langle C \rangle, \langle d \rangle$ Kodierungslängen der pmatrix C bzw. des Vektors d)
- c) $r = \sqrt{n} \cdot 2^{\langle C, d \rangle - n^2}$, $A_0 = 2rE_n$, $a_0 = 0$.

Iteration:

- d) Wenn $k = N$, dann STOP, Ausgabe: $P = \emptyset$.
- e) $a_k \in P$, a_k Zentrum von $E(A_k, a_k)$, so STOP, Ausgabe: $P \neq \emptyset$.
- f) $a_k \notin P$, wähle Ungleichung $c_i^T x \leq d_i$, die durch a_k verletzt wird. Setze $c := c_i$.

Definiere:

- g) $b := \frac{1}{\sqrt{c^T A_k c}} A_k c$
- h) $a_{k+1} := a_k - \frac{1}{n+1} b$
- i) $A_{k+1} := \frac{n^2}{n^2-1} (A_k - \frac{2}{n+1} b b^T)$
- j) Gehe zu d)

Satz 12.3. *Der Ellipsoid-Alg. terminiert nach $\mathcal{O}(mn^2 \langle C, d \rangle)$ arithmetischen Operationen und gibt entweder $x \in P$ oder $P = \emptyset$ aus.*

Proposition 12.4.

$$\text{vol}(E(A, a)) = \sqrt{\det A} \cdot \tau_n.$$

τ_n ist das Volumen der Einheitskugel im \mathbb{R}^n .

$$\tau_n = \frac{\pi^{\frac{n}{2}}}{\Gamma(\frac{n}{2} + 1)} \sim \frac{1}{\sqrt{\pi n}} \cdot \left(\frac{2e\pi}{n}\right)^{\frac{n}{2}}.$$

Proposition 12.5. *Sei $T: x \mapsto Dx + d$ eine bijektive, affine Abbildung. Dann gilt*

(i)

$$\text{vol}(T(E(A, a))) = \det D \cdot \text{vol}(E(A, a)).$$

(ii)

$$\frac{\text{vol}(E(A, a))}{\text{vol}(E(A', a'))} = \frac{\text{vol}(T(E(A, a)))}{\text{vol}(T(E(A', a')))}.$$

Lemma 12.6 (Initialisierungslemma). Sei $P = \{x \in \mathbb{R}^n \mid Cx \leq d\}$ mit $C \in \mathbb{Z}^{m \times n}$, $d \in \mathbb{Z}^m$. Sei $r = \sqrt{n} \cdot 2^{\langle C, d \rangle - n^2}$. Dann gilt:

(i) $\text{ex}(P) \neq \emptyset$, so ist $\text{ex}(P) \subseteq B(0, r)$.

(ii) Ist P ein Polytop, so gilt $P \subseteq B(0, r)$.

Lemma 12.7 (Abbruchkriterium). Ist P volldimensional, dann gilt $\text{vol}(P) \geq 2^{-(n+1)\langle C, d \rangle + n^3}$.

Beweis des Satzes 12.3. $N = \# \text{Iteration} = 2n((2n+1)\langle C \rangle + n\langle d \rangle - n^3)$. Nach Shrinking-Lemma (12.1) verkleinert sich das Volumen um einen Faktor von $e^{-\frac{1}{2(n+1)}}$. Nach N Iterationen ist das Ausgangsvolumen ($= r^n \tau_n$) kleiner als $2^{-(n+1)\langle C, d \rangle + n^3}$. Mit Lemma 12.7 folgt: Der Algorithmus muss dann, oder schon vorher terminiert haben.

Laufzeit:

- Initialisierung kostet $\mathcal{O}(mn)$, da zur Berechnung von $\langle C \rangle$ bzw. $\langle C, d \rangle$ mn pmatrixeinträge zu betrachten sind.
- Jede Iteration kostet $\mathcal{O}(mn)$, da zur Berechnung, ob $a_k \in P$ m Ungleichungen getestet werden und jeder Test kostet $\mathcal{O}(n)$. (Updates gehen auch in dieser Zeit)
- $\# \text{Iterationen} = N = \mathcal{O}(n^2 \langle C, d \rangle)$

$\Rightarrow \mathcal{O}(mn^3 \langle C, d \rangle)$ Gesamtkosten. □

Beweisidee für Lemma 12.6. (i) Zunächst sind sie Lösung eines Teilsystems $C'v = d'$ von $Cx = d$. Mit der Cramer'schen Regel können wir die Lösung v schreiben als:

$$v_i = \frac{\det C'_i}{\det C'}$$

C'_i Untermatrix von C .

$$v \in \text{ex}(P) \Leftrightarrow \exists C', d' \text{ so dass } C'v = d' \text{ endlich bestimmt}$$

Nun z.Z.

$$\|v\| = \sqrt{v_1^2 + \dots + v_n^2} \leq r.$$

$\det(C) \leq \text{Hadamondmatrix}$

(ii) mit Krein-Milman: $\text{co}(\text{ex}(P)) = P$.

□

Beweisidee für Lemma 12.7. P voll-dimensional $\Rightarrow P$ hat $n+1$ affin-unabhängige Ecken v_0, \dots, v_n .

$S := \text{co}(\{v_0, \dots, v_n\})$ ist ein Simplex. $S \subseteq P$, also $\text{vol}(P) \geq \text{vol}(S)$.

Analysis III:

$$\text{vol}(S) = \frac{1}{n!} \left| \det \begin{pmatrix} 1 & \dots & 1 \\ v_0 & \dots & v_n \end{pmatrix} \right|.$$

□

13 Travelling-Salesman-Problem

Definition 13.1 (TSP – symmetrischer Fall). Eine Instanz des *TS-Psymm.* ist gegeben durch ein $n \in \mathbb{N}$, den Graphen $G = K_n$ mit Kantenmenge E und Knotenmenge $V = \{1, \dots, n\}$, sowie einer Kosten- oder Distanzfunktion $c: E \rightarrow \mathbb{R}^+$ (symmetrisch, da G ungerichtet).

Sei $T = v_1, \dots, v_k$ ein Kreis in K_n (mit v_1, \dots, v_k paarweise verschieden). Die Länge bzw. Kosten von T sind

$$c(T) := \sum_{i=1}^{k-1} c(\{v_i, v_{i+1}\}) + c(\{v_k, v_1\})$$

Wir betrachten im folgenden *Hamiltonkreise* T , d.h. solche Kreise, die jeden Knoten *genau* einmal besuchen. Solche T heißen auch hier *TSP-Tour* oder *Rundreise*.

Gesucht ist eine günstige TSP-Tour, d.h. eine kostenminimale TSP-Tour (Existenz: Es existiert eine TSP-Tour und endlich viele verschiedene Touren). Die Länge einer günstigen bzw. kürzesten Tour bezeichnen wir als OPT.

Beachte: Es existieren exponentiell viele Touren (nämlich sogar $(n-1)!$ viele), durch reines kombinatorisches Ausprobieren lässt dich also keine kürzeste Tour effizient ermitteln!

Definition 13.2 (TSP – allgemeiner Fall). Betrachten wir $G = \overrightarrow{K}_n$, den gerichteten vollständigen Graphen, so erhalten wir das *allgemeine TSP*. Im Folgenden soll allerdings nur der symmetrische Fall (mit Dreiecksungleichung) betrachtet werden.

Definition 13.3 (TSP – mit Dreiecksungleichung). Wir betrachten als Subproblem zu TSPsymm. das *TSP mit Dreiecksungleichung (TSP Δ -Ungl.)*, d.h. es gilt:

$$c(\{v, w\}) \leq c(\{v, u\}) + c(\{u, w\})$$

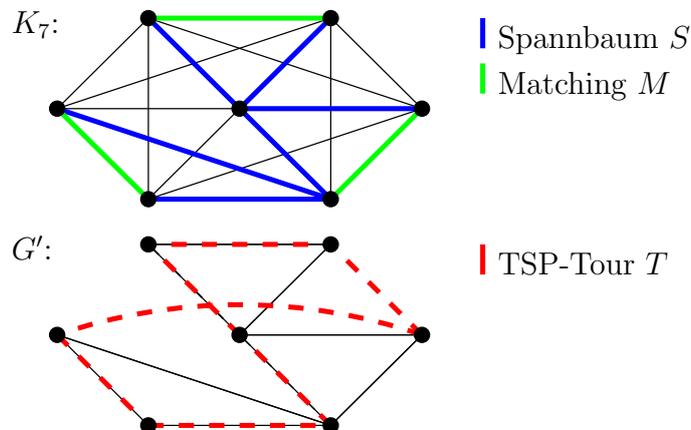
für alle $u, v, w \in V$.

13.1 Algorithmus von Christofides

Algorithmus von Christofides für TSP Δ -Ungl.:

- (1) Berechne (*) MST (minimum spanning tree) S in K_n bezüglich c .
- (2) (a) Setze $V' := \{v \in V \mid \deg_S(v) \text{ ungerade}\}$. (beachte: $|V'|$ gerade)
 - (b) Berechne (**) ein perfektes Matching M mit minimalen Kosten zwischen den Knoten aus V' in K_n bezüglich c .
 - (c) Setze $G' := (V, E(S) \cup M)$. (beachte: G' kann Multigraph sein!)
- (3) Berechne (***) eine Euler-Tour R in G' , d.h. einen Kreis R , der jede Kante in $E' := E(S) \cup M$ genau einmal durchläuft.
- (4) Gewinne aus R durch Entfernen innerer Kreise (shortcut-Technik) eine TSP-Tour T .

Beispiel.



Satz 13.4. *Der Algorithmus von Christofides terminiert in $\mathcal{O}(n^3)$ Zeit und liefert eine TSP-Tour T mit Approximationsgüte $\frac{3}{2}$, d.h. $c(T) \leq \frac{3}{2} \cdot \text{OPT}$.*

Beweis. (a) zu Terminierung und Laufzeit: Es gilt $|V'| = 2k$ für ein $k \in \mathbb{N}$ (Anzahl der Knoten ungeraden Grades in Bäumen ist gerade), da:

$$|E(S)| = \frac{1}{2} \underbrace{\sum_{v \in V(S)} \deg_S(v)}_{\text{gerade}} \in \mathbb{N}.$$

Also existiert ein perfektes Matching der Knoten aus V' in K_n . Kostenminimales perfektes Matching kann etwa mit dem Algorithmus von Edmonds (***) in $\mathcal{O}(n^3)$ berechnet werden (2.b).

Den MST in (1) finden wir etwa mit dem Algorithmus von Kruskal (*) in $\mathcal{O}(nm) = \mathcal{O}(n^3)$.

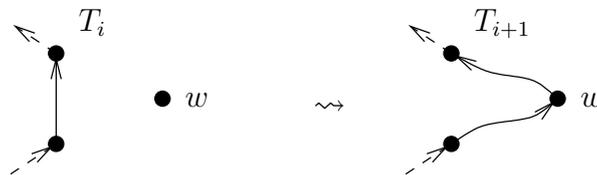
Alle weiteren Schritten sind in $\mathcal{O}(n^2)$. Damit erhalten wir eine Laufzeit von $\mathcal{O}(n^3)$, und der Algorithmus terminiert.

- (b) zur Approximationsgüte: Sei T_{OPT} eine optimale TSP-Tour, also: $c(T_{\text{OPT}}) = \text{OPT}$. Entfernen einer Kante aus T_{OPT} liefert einen spannenden Baum S_{OPT} mit Kosten $c(S_{\text{OPT}}) \leq c(T_{\text{OPT}}) = \text{OPT}$. Also gilt für den MST S : $c(S) \leq c(S_{\text{OPT}}) \leq \text{OPT}$. T_{OPT} besucht alle Knoten, insbesondere diejenigen aus V' . Finde Pfade p_1, \dots, p_{2k} und v_1, \dots, v_{2k} so, dass $T_{\text{OPT}} = v_1 p_1 \dots v_{2k} p_{2k}$ (*) und $V' = \{v_1, \dots, v_{2k}\}$. $M_1 := \{\{v_1, v_2\}, \dots, \{v_{2k-1}, v_{2k}\}\}$ und $M_2 := \{\{v_2, v_3\}, \dots, \{v_{2k}, v_1\}\}$ sind zwei perfekte Matchings von V' . Entfernt man in T_{OPT} (*) die Pfade p_1, \dots, p_{2k} , so entsteht ein Kreis, der abwechselnd M_1 und M_2 durchläuft. Wegen der Dreiecksungleichung für TSP Δ -Ungl. gilt: $c(M_1) + c(M_2) \leq c(T_{\text{OPT}}) = \text{OPT}$, also o.B.d.A. $c(M_1) \leq \frac{1}{2}\text{OPT}$. Da M kostenminimal ist, folgt: $c(M) \leq c(M_1) \leq \frac{1}{2}\text{OPT}$.

Es gilt nun für T und die Euler-Tour R nach Knostruktion und Dreiecksungleichung: $c(T) \leq c(R) = c(M) + c(S) \leq \frac{1}{2}\text{OPT} + \text{OPT} = \frac{3}{2}\text{OPT}$. \square

13.2 Insert-Heuristiken

Idee: Konstruiere eine Folge von Subtours T_1, \dots, T_n , wobei T_{i+1} immer genau einen Knoten mehr als T_i besucht. Es wird in jedem Schritt nach einer Heuristik der als nächstes einzufügende Knoten w gewählt. Steht dieser fest, so wird w möglichst kostengünstig in die bestehende Subtour eingefügt.



Bezeichne den Kostenzuwachs durch das Einfügen von w als

$$\Delta c(T_i, w) := \min \{c(T_i) - c(\{u, v\}) + c(\{u, w\}) + c(\{w, v\}) \mid \{u, v\} \in T_i\}.$$

Wie wird w bestimmt?

- (a) **nearest-Insert:** Wähle w so, dass $d(T_i, w) := \min \{c(\{v, w\}) \mid v \in T_i\}$ minimiert wird. Bewiesene Güte: 2.
- (b) **farthest-Insert:** Wähle w so, dass $d(T_i, w)$ maximiert wird. (Bisher keine bewiesene Güte, aber mindestens 2)
- (c) **cheapest-Insert:** Wähle w so, dass $\Delta c(T_i, w)$ minimiert wird. Bewiesene Güte: 2.

In der Praxis hat sich farthest-Insert als effektivste Heuristik herausgestellt, allerdings unbewiesen!

(d) alternativ: **Verbesserungsheuristiken**

- k -OPT, wobei $k = 2$ oder $k = 3$, üblicherweise.
- weitere: Lin und Kernighan.

13.3 Approximationsschema von Arora

Definition 13.5 (Partitionsbäume). (i) Das kleinste Quadrat mit Nullpunkt als linke untere Ecke, welche die TSP-Instanz $I \subseteq \mathbb{R}_{\geq 0}^2$ enthält und eine Seitenlänge hat, die eine 2-er Potenz ist, heißt *Bounding Box*. Die wird mit $R(I)$ bezeichnet.

(ii) Ihre Seitenlänge bezeichnen wir mit $L = L(I) = 2^l$.

Wir zerlegen R in 4 kleinere Quadrate und führen den Prozeß solange durch, bis die Seitenlänge der Quadrate ≤ 1 ist. Da L 2-er Potenz, haben die kleinsten Quadrate Seitenlänge genau 1.

Definition 13.6. Eine *vertikale Gitterlinie* ist eine Menge in \mathbb{R}^2 , für die es ein $i \in \mathbb{N}_0$ gibt, so dass ihre Elemente alle x -Koordinate i haben. Analog ist eine *horizontale Gitterlinie* definiert.

Die Menge der vertikalen bzw. horizontalen Gitterlinien bezeichnen wir mit \mathcal{L}_v bzw. \mathcal{L}_h , und setzen $\mathcal{L} = \mathcal{L}_v \cup \mathcal{L}_h$.

Für $k \in \mathbb{N}$, $a, b \in \mathbb{N}_0$ sei $\mathcal{L}(k, a, b)$ die Menge aller vertikalen Gitterlinien mit x -Koordinate der Form $kj + a$ und aller horizontalen Gitterlinien mit y -Koordinate der Form $kj + b$, $j \in \mathbb{N}_0$.

$\mathcal{L} = \mathcal{L}(1, 0, 0)$.

$\mathcal{L}(k, a, b)$ ist ein um (a, b) verschobenes Gitter mit Gitterlinienabstand k .

O.B.d.A liege kein Punkt aus I auf einer Gitterlinie.

Definition 13.7. Ein *Partitionsbaum* von R ist ein Quadrupel (Q, E, q_0, ϕ) , wobei (Q, E) ein Baum ist. $q_0 \in Q$ und $\phi: Q \rightarrow \mathcal{P}(R)$ mit den Eigenschaften:

- a) $\phi(q_0) = R$
- b) Es gilt $(q, q') \in E \Leftrightarrow \phi(q) \supseteq \phi(q')$ und für alle $p \in Q$ aus $\phi(p) \supset \phi(q')$ schon $\phi(p) \supseteq \phi(q)$ folgt.

Es gibt $a, b \in \mathbb{N}_0$, so dass gilt

- c) Ordnen wir Q von q_0 als Wurzel ausgehend in Ebenen an, so gilt:

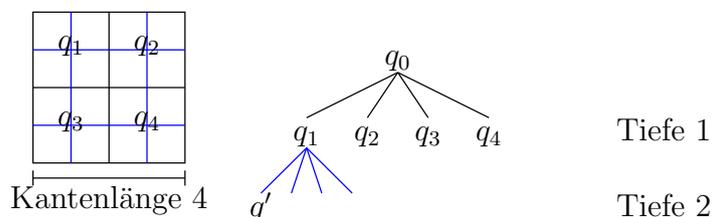
$$\{\phi(q) \mid q \text{ ist auf Ebene } i\}$$

ist eine Prartition von $R \setminus \mathcal{L}(2^{l-i}, a, b)$.

- d) für alle $q \in Q$ auf Ebene i besteht $\phi(q)$ aus der Vereinigung von einer oder mehrerer Zusammenhangskomponenten von $R \setminus \mathcal{L}(2^{l-i}, a, b)$. Wir nennen $\phi(q)$ auch den Bereich von q und $\{\phi(q) \mid q \in Q\}$ die Menge der Bereiche des Partitionsbaums.

- e) Ein *kanonischer Partitionsbaum* hat die Eigenschaft: $\phi(q)$ ist ein Quadrat für alle $q \in Q$.

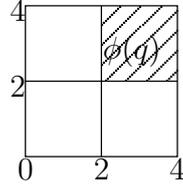
Beispiel zur Def.:



Definition 13.8 ((a, b)-Shifts). Seien $a, b \in [0, L] \cap \mathbb{Z}$. Unter dem (a, b) -verschobenen Partitionsbaum $P_{a,b}$ verstehen wir einen Partitionsbaum, der aus dem kanonischen durch Veränderung von ϕ zu $\phi_{a,b}$ wie folgt hervorgeht: Für jedes $q \in Q$ definieren wir

$$Q_{a,b}(q) := \left\{ \left(\underbrace{a+x \pmod L}_{x'}, \underbrace{b+y \pmod L}_{y'} \right) \mid (x, y) \in \phi(q) \right\}$$

Beispiel: $L = 4 = 2^2$, $a = 0$, $b = 2$. $\phi_{0,2}(q) = \{x \bmod 4, 2 + y \bmod 4 \mid (x, y) \in \phi(q)\}$



Proposition 13.9. $P_{a,b}$ ist wieder ein Partitionsbaum und $P_{0,0} = P$.

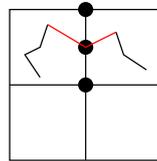
O.B.d.A.: Abstand zwischen $i, j \in I$ soll mind. 8 sein und jedes Blatt des Partitionsbaums enthält höchstens ein $i \in I$.

Definition 13.10. Der Grad einer Gitterlinie l bezeichnet man mit $\deg(l)$. Er ist bezüglich eines Partitionsbaumes die minimale Ebene aller angrenzenden Bereiche.

Definition 13.11. Ein Salesman-Pfad ist ein geschlossener Polygonzug in \mathbb{R}^2 , der alle Punkte von I besucht.

Definition 13.12. Ist $m \in \mathbb{N}$ und $P_{a,b}$ ein Partitionsbaum, so bezeichnen wir mit einer m -regulären Menge von Portalen für $P_{a,b}$ eine m elementige Menge von Punkten, die alle auf den Seiten der Quadrate liegen, jedes Quadrat in seinen Ecken ein Portal hat, und diese gleichmäßig verteilt sind.

Beispiel zur Def.:



Definition 13.13. Seien $m, r \in \mathbb{N}$, $P_{a,b}$ und eine m -reguläre Menge von Portalen für $P_{a,b}$ gegeben. Ein Salesman-Pfad heißt (m, r) -leicht, wenn er die Seiten eines jeden Bereiches von $P_{a,b}$ höchstens r mal passiert und das stets nur in einem Portal.

Satz 13.14. Seien $a, b \in [0, L) \cap \mathbb{Z}$ zufällig gewählt. Dann gilt mit W keit mindestens $\frac{1}{2}$: Es gibt einen (m, r) -leichten Salesman-Pfad, wobei $m = \mathcal{O}(c \log L)$, $r = \mathcal{O}(c)$, $c > 0$ eine bel. Konstante, dessen Länge höchstens $(1 + \frac{1}{c})\text{OPT}$ ist.

Lemma 13.15 (Patching Lemma). Es gibt eine Konstante $g > 0$, so dass folgendes gilt: Ist S eine Strecke der Länge s und ist π eine geschlossene Kurve, die S wenigstens dreimal kreuzt. Dann findet man Teilstücke von S mit Gesamtlänge höchstens gs , deren Hinzufügen zu π eine geschlossene Kurve π' ergibt, die S höchstens zweimal schneidet.

Definition 13.16. Sei π ein Polygonzug in \mathbb{R}^2 und $l \in \mathcal{L}$ eine Gitterlinie. Dann bezeichne $t(\pi, l)$ wie oft π die Gitterlinie l schneidet.

Lemma 13.17. *Sei π eine optimale TSP-Tour der Länge T . Dann gilt:*

$$\sum_{l \in \mathcal{L}_v} t(\pi, l) + \sum_{l \in \mathcal{L}_h} t(\pi, l) \leq 2T.$$

Generalannahme: Punkte aus I mindestens 4 voneinander entfernt.

Beweis des Struktursatzes. Sei g die Konstante aus Lemma 13.12, sei $s := \max\{12gc, 15\}$, $r := s + 8$, wähle $m \geq 2s \log L$. Sei π eine optimale TSP-Tour. Wir ändern π so ab, dass sie bezüglich $P_{a,b}$, mit gewissen a, b , ein (m, r) -leichter Salesmanpfad wird. Dazu:

1. Schritt: Reduziere Anzahl der Kreuzungen von π an Gitterlinien (mit dem Patchinglemma).
2. Schritt: Erzwingen Durchgang nur durch Portale.

Wir haben L Gitterlinien sowie 2^i vom Grad i . Für eine Gitterlinie l vom Grad $i = 1, \dots, \log L$, gilt:

$$\Pr[\deg(l) = i] = \frac{2^i}{L} \tag{1}$$

Bei einer Gitterlinie vom Grad i betrachte die Teilstücke zwischen den y -Koordinaten $b + p \frac{L}{2^i}$ und $b + (p + 1) \frac{L}{2^i}$, $p = 0, \dots, 2^i - 1$.

Algorithmus Modify

Eingabe: Gitterlinie l vom Grad i .

Ausgabe: Modifizierter Salesmanpfad.

Von $j = \log L$ bis $j = i$ führe aus:

Von $p = 0$ bis $p = 2^i - 1$ führe aus:

Falls das Liniensegment zwischen den y -Koordinaten

$b + p \frac{L}{2^i}$ und $b + (p + 1) \frac{L}{2^i}$

von π mehr als s mal ueberschritten wird, so

reduziere die Anzahl der Schnitt mit Patching-Lemma zu 4.

Fragen:

- a) Um wieviel verlängert Modify die Tour?
- b) Wie wirkt sich die Modifikation auf die Gesamtzahl der Überschreitungen aus?

zu a) Sei $\Delta_1(l, i)$ der Längenzuwachs, der durch Anwenden von Modify mit Eingabe l und i entsteht. Sei $c_{l,j}$ die Anzahl der Anwendungen des Patching-Lemma in Modify auf l in Iteration j der äusseren For-Schleife.

$$\sum_{j \geq 1} c_{l,j} \leq \frac{t(\pi, l)}{s-3}, \quad \Delta(l, i) \leq \sum_{j \geq i} c_{l,j} g \frac{L}{2^j} \quad (2)$$

$$\begin{aligned} \mathbb{E}(\Delta_1(l, \deg(l))) &= \sum_{i \geq 1} \underbrace{\Pr[\deg(l) = i]}_{=\frac{2^i}{L}, \text{Claim1}} \cdot \underbrace{\Delta_1(l, i)}_{\text{Claim2}} \leq \sum_{i \geq 1} \frac{2^i}{L} \cdot \sum_{j \geq 1} c_{l,j} g \frac{L}{2^j} \\ &\leq g \sum_{i \geq 1} \sum_{j \geq i} c_{l,j} 2^{i-j} = g \sum_j c_{l,j} \sum_{i \leq j} 2^{i-j} \\ &\leq g \sum_{j \geq 1} c_{l,j} \underbrace{\sum_{i=0}^{j-1} \left(\frac{1}{2}\right)^i}_{\leq 2, \text{geom. Reihe}} \leq 2g \sum_{j \geq 1} c_{l,j} \stackrel{\text{Claim2}}{\leq} 2g \cdot \frac{t(\pi, l)}{s-3} \end{aligned}$$

Also

$$\sum_{l \in \mathcal{L}} \mathbb{E}(\Delta_1(l, \deg(l))) \stackrel{\text{Lemma13.14}}{\leq} \frac{4g}{s-3} \cdot T.$$

zu b) Sei $\Delta_2(l, i)$ der Längenzuwachs, der durch Schritt 2 an Gitterlinie l vom Grad i entsteht. Es gilt

$$\mathbb{E}(\Delta_2(l, \deg(l))) \leq \frac{t(\pi, l)}{2s}.$$

Damit:

$$\mathbb{E}(\Delta l) = \mathbb{E}(\Delta_1 l) + \mathbb{E}(\Delta_2 l) \leq 2g \frac{t(\pi, l)}{s-3} + \frac{t(\pi, l)}{2s} \leq \frac{3gt(\pi, l)}{s}$$

Sei $\Delta\pi$ der Gesamtlängenzuwachs:

$$\begin{aligned} \mathbb{E}(\Delta\pi) &= \sum_{l \in \mathcal{L}} \mathbb{E}(\Delta l) = \sum_{l \in \mathcal{L}_h} \mathbb{E}(\Delta l) + \sum_{l \in \mathcal{L}_v} \mathbb{E}(\Delta l) \\ &\leq \frac{3g}{s} \left(\underbrace{\sum_{l \in \mathcal{L}_h} t(\pi, l) + \sum_{l \in \mathcal{L}_v} t(\pi, l)}_{\leq 2T, \text{Lemma13.14}} \right) \leq \frac{6g}{s} \cdot T = \frac{6g}{s} \cdot \text{OPT} \end{aligned}$$

Markow-Ungleichung:

$$\Pr[X \geq t] \leq \frac{\mathbb{E}(X)}{t}.$$

Mit $t = 2\mathbb{E}(X) \Rightarrow \Pr[X > \mathbb{E}(X)] \leq \frac{1}{2}$. □

Satz 13.18. Die Konstruktion einer optimalen (m, r) -leichten Salesmantour kann in Zeit $\mathcal{O}(N(m+2)^{8r}(4r)^{4r}(4r)!^2)$, wobei N die # des in q verwurzelten Teilbaumes des Partitionsbaumes ist, durchgeführt werden.

Satz 13.19 (Arora 1996). Mit W'keit mindestens $\frac{1}{2}$ kann eine TSP-Tour der Länge $\leq (1 + \epsilon)\text{OPT}$ konstruiert werden, in Laufzeit $\mathcal{O}(n^{\mathcal{O}(\frac{1}{\epsilon})})$.

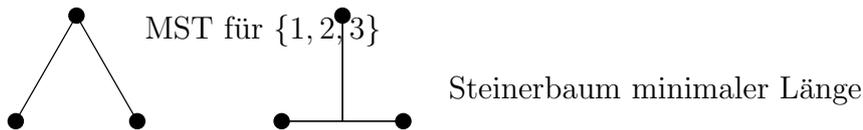
14 Steinerbäume

Steinerbäume sind Netzwerke, die eine gegebene Punktmenge längen-minimal verbinden.

Definition (SP in \mathbb{R}^d). Gegeben $P \subseteq \mathbb{R}^d$, $|P| < \infty$, sowie $S \subseteq \mathbb{R}^d$, $|S| < \infty$. Ein *Steinerbaum* T für P mit *Steinerpunktmenge* S ist ein Baum in \mathbb{R}^d für $P \cup S$.

Problem (SP in \mathbb{R}^d). Gegeben P , finde S , so dass der Steinerbaum für P mit Steinermenge S minimale Länge hat.

Beispiel.



SP in Graphen:

Definition 14.1. Sei $G = (V, E)$ ein Graph mit ausgezeichnete Menge $T \subseteq V$, sogenannte *Terminalmenge*. Ein Subgraph $B = (V(B), E(B))$ von G heisst *Steinerbaum* für T , falls

- (i) B ist ein Baum in G mit $T \subseteq V(B)$.
- (ii) Jedes Blatt von B ist ein *Terminal*, d.h. aus T .
- (iii) Die Elemente von $S = V(B) \setminus T$ heissen *Steinerknoten*.

Sei $l: E \rightarrow \mathbb{Q}_0^+$ eine *Längen- oder Kostenfunktion*. Für $E' \subseteq E$ sei $l(E') = \sum_{e \in E'} l(e)$ die sogenannte *Länge* von E' .

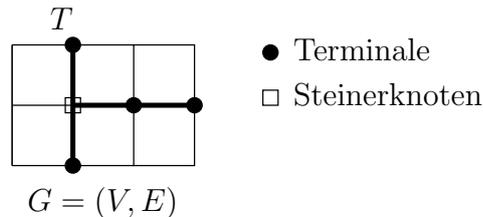
Ein Steinerbaum B für T mit minimaler Länge heisst *steinerminimaler Baum*.

Problem 14.2 (SPG). *Eingabe:* $G = (V, E)$, $l, T \subseteq V$, $k \in \mathbb{Q}_0^+$.

Frage: Gibt es einen Steinerbaum B für T mit $l(B) \leq k$?

Zugehöriges Optimierungsproblem besteht darin einen steinerminimalen Baum für T zu berechnen.

Beispiel. •



- VLSI-Design

Logikchip:

- ca. 1 Mio – 10 Mio Transistoren
- ca. 5 Mio – 50 Mio Netze
- quasioptimale Verdrahtungen haben Gesamtlänge von 15 – 60m
- Taktzeiten ca. 1 – 3ns

Entwicklungszeiten:

Physical Layout ca. 5 – 6 Monate oder 1 Jahr

Logical Layout, Test mehr als 6 Monate

[Bedeutung der Steinerbaum-Probleme liegt im VLSI-Design]

Satz 14.3. *SPG und SP in \mathbb{R}^d ist NP-vollständig.*

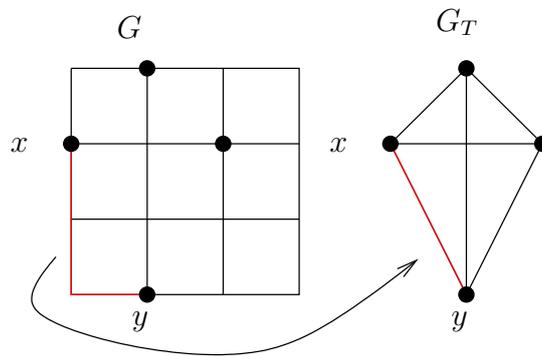
Sei $\rho = \frac{\text{Länge MST}}{\text{Länge steinermin. Baum}}$.

- a) Vermutung $\rho(\mathbb{R}^2) = \frac{2}{\sqrt{3}}$. (Gilbert, Pollack 1966)
Bewiesen durch Du, Hwang (1992)
- b) $\rho(\text{Graphen}) < 2$. In den letzten 15 Jahren große Fortschritte hin zu $\rho(\text{Graphen}) < 1.5$.
Untere Schranke der Nicht-Approximierbarkeit: $\rho \geq 1.0074$ (unter Annahme der Form $P \neq NP$)

KMB-Algorithmus (Kou, Markowsky, Berman)

Definition 14.4. Sei $G = (V, E)$ ein Graph und $T \subseteq V$, sowie $l: E \rightarrow \mathbb{Q}_0^+$ gegeben. Der *Distanzgraph* G_T für T ist der vollständige Graph über T , wobei die Kanten (x, y) , $x, y \in T$, die Länge $l_T(x, y)$ haben, und $l_T(x, y)$ ist die Länge des kürzesten Weges von x nach y in G .

Idee: $G \rightarrow G_T \xrightarrow{MST} B_T \xrightarrow{\text{in } G} B_T^* \xrightarrow{\text{Erzeuge Baum}} B$, B ist ein Steinerbaum für T .



Lemma 14.5. Sei B_{OPT} ein steinerminimaler Baum für T in G und sei B_T der MST in G_T . Sei b die Anzahl der Blätter von B_T . Dann gilt $l(B_T) \leq 2(1 - \frac{1}{b})l(B_{\text{OPT}})$.

Beweis. Wir betrachten B_{OPT} hilfswise in der Ebene, etwa: Sei R eine B_{OPT} einschließende Tour, dann $l(R) = 2l(B_{\text{OPT}})$. Sei R^* erzeugt durch Entfernen des längsten Tourstrichs aus R , das zwei Terminale verbindet. Damit: $l(R) - l(R^*) \geq \frac{l(R)}{b}$. Also $l(R^*) \leq l(R) - \frac{l(R)}{b} = 2(1 - \frac{1}{b})l(B_{\text{OPT}})$. R^* induziert einen spannenden Baum B^* für T . Also: $l(B_T) \leq l(B^*) \leq l(R^*)$. \square

Algorithmus KMB

1. Bestimme G_T, l_T
2. Bestimme MST B_T in G_T
3. B_T induziert in G einen Subgraphen B_T^*
4. Bestimme MST B_T^{**} für B_T^*
5. Erzeuge aus B_T^{**} einen Steinerbaum B_{KMB} durch sukzessives Löschen von Kanten, die zu Blättern gehören, die keine Terminale sind.

Beispiel.

Satz 14.6. Der KMS-Algorithmus berechnet B_{KMB} in Zeit $\mathcal{O}(|T|n^2)$ und es gilt $l(B_{\text{KMB}}) \leq 2(1 - \frac{1}{b})l(B_{\text{OPT}})$, ($b = \text{Anzahl der Blätter in } B_{\text{OPT}}$)

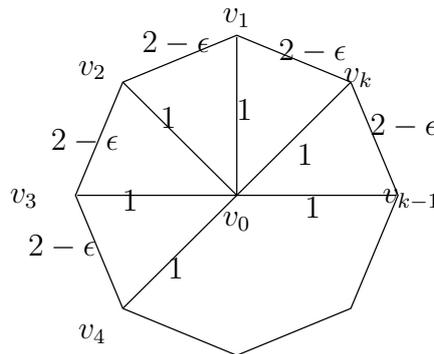
Beweis. Gütegarantie liefert Lemma 14.5.

Laufzeit:

1. $\mathcal{O}(|T|n^2)$
2. $\mathcal{O}(|T|^2)$
3. $\mathcal{O}(n)$
4. $\mathcal{O}(n^2)$
5. $\mathcal{O}(n)$

□

Beispiel (2-Approximation für MST/Steinerbaum ist scharf).



$$G = (V, E, l)$$

$$T = V \setminus \{v_0\}$$

$$\text{Also: } l(\text{MST}(T)) = (2 - \epsilon)(k - 1)$$

$$l(\text{SMT}(T)) = k$$

Also

$$\frac{l(\text{MST}(T))}{l(\text{SMT}(T))} = (2 - \epsilon) \frac{k - 1}{k} = (2 - \epsilon) \left(1 - \frac{1}{k}\right) \xrightarrow{\epsilon \rightarrow 0, k \rightarrow \infty} 2$$

Exkurs: Hypergraphen

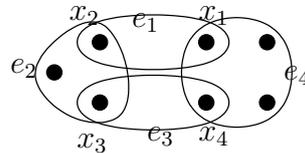
Ein *Hypergraph* $\mathcal{H} = (V, \mathcal{E})$ besteht aus einer endlichen Menge V und $\mathcal{E} \subseteq \mathcal{P}(V)$. Die Elemente aus \mathcal{E} heissen *Hyperkanten*.

\mathcal{H} ist ein Graph, gdw. $|E| = 2$ für alle $E \in \mathcal{E}$.

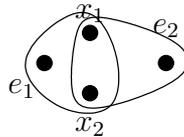
Ein u - v -Pfad in \mathcal{H} der Länge $l \geq 2$ ist eine Folge $x_1, e_1, \dots, x_l, e_l, x_{l+1}$ von Knoten $x_i \in V$ mit $x_1 = u$ und $x_{l+1} = v$ und Kanten $e_i \in \mathcal{E}$, so dass die x_i verschieden sind, die e_i verschieden sind und $x_1 \in e_1, x_{l+1} \in e_l, x_i \in e_{i-1} \cap e_i$ für alle $i = 2, \dots, l$.

Ein Kreis der Länge $l \geq 2$ in \mathcal{H} ist ein Pfad $x_1, e_1, \dots, x_{l-1}, e_{l-1}, x_l$ mit der zusätzlichen Eigenschaft, dass x_1 und x_l in einer Kante e_l enthalten sind, die verschieden ist von e_1, \dots, e_{l-1} .

Beispiel. Kreis der Länge 4:



Kreis der Länge 2:



In Graphen ist kleinster Kreis der K_3 , während es in Hypergraphen Kreise der Länge 2 gibt.

Ein Hypergraph \mathcal{H} ist *zusammenhängend*, wenn für jedes Paar $u, v \in V$ ein u - v -Pfad in \mathcal{H} existiert.

Ein Hypergraph \mathcal{H} ist ein *Baum*, wenn er zusammenhängend ist und keinen Kreis enthält.

Ein *aufspannender Baum* ist ein Subhypergraph \mathcal{G} von \mathcal{H} mit $\mathcal{G} = (V(\mathcal{G}), \mathcal{E}(\mathcal{G}))$, so dass $V(\mathcal{G}) = V$ und \mathcal{G} ein Baum ist.

Jeder zusammenhängende Graph enthält einen aufspannenden Baum, aber ein Hypergraph, der ein 2-Kreis ist, enthält keinen aufspannenden Baum.

Ein r -*uniformer* Hypergraph besitzt nur Hyperkanten der Kardinalität r .
(\rightsquigarrow Ein Graph ist ein 2-uniformer Hypergraph)

Ein Hypergraph ist k -*regulär*, wenn in jedem Knoten genau k Hyperkanten inzident sind.

Ein r -*beschränkter* Hypergraph besitzt nur Kanten der Kardinalität $\leq r$.

Sei $w: \mathcal{E} \rightarrow \mathbb{R}$. Für $\mathcal{E}' \subseteq \mathcal{E}$ ist $w(\mathcal{E}') := \sum_{E \in \mathcal{E}'} w(E)$ das *Gewicht* von \mathcal{E}' .

$\text{mst}(\mathcal{H})$ ist das Gewicht eines aufspannenden Baumes in \mathcal{H} mit minimalem Gewicht, fall existent. Falls kein aufspannender Baum existiert, setzen wir $\text{mst}(\mathcal{H}) = \infty$.

Sei $G = (V, E)$, l Längenfunktion und $K \subseteq V$ Terminalmenge eine Instanz des Steinerbaumproblems.

Definiere einen r -beschränkten, $r \geq 2$, Hypergraphen $\mathcal{H}_r(K, \mathcal{E}_r, l_r)$: \mathcal{E}_r ist die Menge aller Teilmengen von K der Größe $\leq r$. Für eine Kante $E \in \mathcal{E}_r$ sei $l_r(E) := \text{smt}(E)$ (in G ; smt ist die Länge des minimalen Steinerbaumes).

$\mathcal{H}_r = \mathcal{H}_r(G)$ heisst der r -Distanzhypergraph von G .

Es gilt: $\mathcal{H}_2(G) = G_D$ der Distanzgraph von G .

Wir schreiben: $\text{mst}_r(G) := \text{mst}(\mathcal{H}_r(G))$.

Proposition 14.7. *Es gilt $\text{mst}_r(G) \geq \text{smt}(G)$ für alle r .*

Beweis. Sei T ein MST in $\mathcal{H}_r(G)$. Für jede Kante E in T sei T_E ein steinerminimaler Baum für E in G . Da T spannender Baum in $\mathcal{H}_r(G)$ ist, gilt: $S = \bigcup_{E \text{ Kante in } T} T_E$ ist ein Subgraph von G , zusammenhängend, der alle Knoten von K (Terminalmenge von G) enthält.

Also: $\text{mst}_r(G) \geq \text{smt}(G)$. □

Frage: Kann man Proposition 14.7 auch in umgekehrter Richtung beweisen, d.h. gilt $\text{mst}_r(G) \leq \alpha \cdot \text{smt}(G)$, $\alpha > 0$?

Proposition 14.8. *Sei r eine feste Konstante > 0 . Gegeben G, l und K , der Hypergraph $\mathcal{H}_r(G)$ lässt sich in Zeit $\mathcal{O}(n^2 \log n + nm + k^{r+1}n^2)$ berechnen, $k = |K|$. Ist T ein spannender Baum von $\mathcal{H}_r(G)$, dann lässt sich ein Steinerbaum S für K in G mit $l(S) \leq l_r(T)$ in $\mathcal{O}(n^3)$ Zeit berechnen.*

Beweis. Anzahl der r -Teilmengen von K ist $\sum_{i=2}^r \binom{k}{i} \leq k^{r+1}$. Für jede dieser Teilmengen berechnet man einen steinerminimalen Baum, z.B. mit einem Enumerationsalgorithmus (Dreyfuß/Wagner-Algorithmus) in Laufzeit $\mathcal{O}(n^2 \log n + nm + k^{r+1}n^2)$. $\mathcal{O}(n^2 \log n + nm)$ wird benötigt, um alle kürzesten Wege zwischen zwei Knoten zu berechnen. $\mathcal{O}(k^{r+1}n^2)$ wird bei der Enumeration, um den Steinerbaum auf k Knoten zu berechnen, benötigt.

Berechne für jedes T_i eines steinerminimalen Baumes S_i $l(S_i) = l_r(T_i)$. Betrachte Subgraphen $S' = \bigcup_{i=1}^s S_i$. Wir wissen, dass S' zusammenhängend ist und es gilt $l(S') \leq l_r(T)$, nach Proposition 14.7.

Konstruiere in S' einen MST und danach einen Steinerbaum S mit $l(S) \leq l_r(T)$. \square

$\rho_r := \sup_{G,l} \frac{\text{mstr}_r(G)}{\text{smt}(G)}$; $G = (V, E)$ Graph, $l: E \rightarrow \mathbb{Q}_0^+$ Längenfunktion.

Nach Proposition 14.7: $\rho_r \geq 1$.

Satz 14.9. Sei $r \geq 2$, $r = 2^s + t$, $0 \leq t \leq 2^s$, $s \in \mathbb{N}$. Dann gilt

$$\rho_r = \frac{(s+1)2^s + t}{s2^s + t}.$$

Beispiel.

r	2	3	4	5	6	7	8	16	32	64
ρ_r	2	1.67	1.5	1.44	1.4	1.36	1.33			

15 Mehrgüterflüsse

15.1 Fraktionale und ganzzahlige Mehrgüterflüsse

Sei $G = (V, E)$ ein Graph, $n = |V|$, $m = |E|$ und $c: E \rightarrow \mathbb{Q}_0^+$ eine Kapazitätsfunktion. Es seien $(s_1, t_1), \dots, (s_k, t_k)$ Quellen-Senken-Paare, $k \in \mathbb{N}$ gegeben, $s_i, t_i \in V$ für alle i .

Man notiert $d_i = (s_i, t_i)$ als *Gut* (commodity), $i = 1, \dots, k$. Sei $D = \{d_i \mid i = 1, \dots, k\}$.

Ferner haben wir eine *Nachfragefunktion* (demand function) $r: D \rightarrow \mathbb{Q}_0^+$.

$r(d_i)$ gibt die Menge des i -ten Guts an, das von s_i nach t_i in G zu transportieren ist.

Sei T die Menge der Quellen und Senken. Sei $H = (T, D, r)$ der sogenannte *Demand-Graph*. Eine Instanz ist durch (G, H) gegeben.

Wir spezifizieren, was Mehrgüterflüsse sind: Für jedes $d \in D$ sei σ_d eine Orientierung der Kanten von G und $N_d = (V, A_d)$ sei das so entstehende Netzwerk. Für $d \in D$ sei $F(d)$ ein s - t -Fluss in G , $d = (s, t)$.

$(F(d))_{d \in D}$ heißt *Mehrgüterfluss*.

$(F(d))_{d \in D}$ heißt *ganzzahliger* (bzw. 0/1) Fluss, wenn jeder der Flüsse $F(d)$ ganzzahlige (bzw. 0/1) Werte auf den Kanten hat.

$(F(d))_{d \in D}$ heißt *zulässig* bezüglich c (bzw. r), wenn die Summe aller Flüsse über einer Kante $e \in E$ höchstens $c(e)$ ist, für alle $e \in E$ (bzw. der Wert des Flusses $F(d)$ mindestens $r(d)$ ist für alle $d \in D$).

Definition 15.1 (Mehrgüterflussprobleme, Fraktional).

- (i) (Demand-Version) Finde einen Mehrgüterfluss für (G, H, c, r) , wenn eine solcher existiert.
- (ii) (Max-Version) Finde einen Mehrgüterfluss für (G, H, c) mit maximalem Wert $v = \sum_{d \in D} v(F(d))$.

Die ganzzahlige (bzw. 0/1) Version der obigen Probleme ist kanonisch.

Komplexität: Die ganzzahlige Demand-Version ist, schon für $k = 2$, NP-vollständig. Die ganzzahlige Max-Version ist NP-schwer.

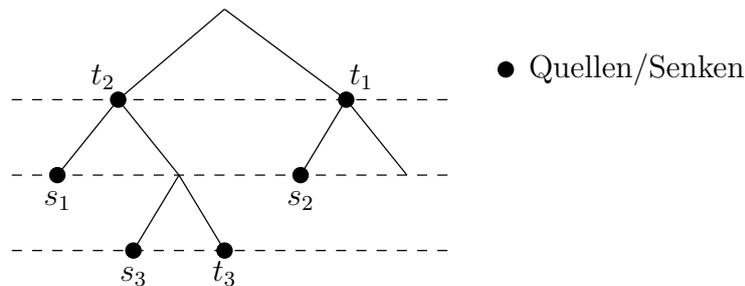
Mit LP lässt sich das fraktionale Problem optimal lösen.

Man kann aus einer LP-Lösung, die uns $F(d)$, $d \in D$ gibt eine Flusspfadmenge konstruieren, $\mathcal{P}(d) = \{P_1, \dots, P_{l(d)}\}$, P_i sind (s, t) -Pfade, wobei auf P_i ein Fluss mit Wert $\lambda(P_i)$ transportiert wird. In der Demand-Version (bzw. Max-Version) gilt $\sum_{P \in \mathcal{P}(d)} \lambda(P) \geq r(d)$ (bzw. $\sum_{d \in D} \sum_{P \in \mathcal{P}(d)} \lambda(P) = \text{OPT(LP)}$).

Man kann zeigen, dass Pfadflussversion \Leftrightarrow LP-Version.

15.2 Ein $\frac{1}{2}$ -Faktor Approximationsalgorithmus für Bäume

Wir betrachten das Max-Problem, wenn $G = (V, E)$ ein Baum ist.



Sei $I = [k]$. Für (s_i, t_i) sei P_i der s_i - t_i -Pfad in G . Sei q_i die charakteristische Funktion von P_i , d.h. $q_i: E \rightarrow \{0, 1\}$, $q_i(e) = 1 \Leftrightarrow e \in P_i$.

Rationales Maximierungsproblem:

$$\begin{aligned}
 (P^*) \\
 \max \sum_{i=1}^k f_i \\
 \sum_{i=1}^k f_i q_i(e) \leq c(e) \text{ für alle } e \in E \\
 f_i \geq 0 \text{ für alle } i \in I
 \end{aligned}$$

(Flussproblem)

Duales Problem:

$$\begin{aligned}
 (D^*) \\
 \min \sum_{e \in E} c(e) d_e \\
 \sum_{e \in E} q_i(e) d_e \geq 1 \text{ für alle } i \in I \\
 d_e \geq 0 \text{ für alle } e \in E
 \end{aligned}$$

(Multicut-Problem)

Seien (P) und (D) die ganzzahligen Versionen von (P^*) und (D^*) . Seien $F_{\text{OPT}}, M_{\text{OPT}}, F^*, M^*$ die Werte der Optima von $(P), (D), (P^*)$ und (D^*) .

Nach dem Dualitätssatz der linearen Programmierung gilt $F^* = M^*$. Also

$$F_{\text{OPT}} \leq F^* = M^* \leq M_{\text{OPT}}.$$

a) Primale Schlupfbedingung: $f_i > 0 \Rightarrow \sum_{e \in P_i} d_e = 1$ für alle $i = 1, \dots, k$.

b) Duale Schlupfbedingung: $d_l > 0 \Rightarrow \sum_{\substack{i=1, \dots, k \\ e \in P_i}} f_i = c(e)$ für alle $e \in E$.

a) und b) sind für (P^*) und (D^*) erfüllt. (Satz vom Komplementären Schlupf).

Was passiert, wenn a) und b) auch für (P) und (D) erfüllt sind?

Seien $f_1, \dots, f_k, d_e, e \in E$ Lösungen von (P) und (D) , die a) und b) erfüllen.

Seien F und M die Werte der Lösungen.

Es gilt: $M = M_{\text{OPT}} = F_{\text{OPT}} = F$.

Sieht man über $M_{\text{OPT}} = F$. ($\Rightarrow F \leq F_{\text{OPT}} \leq M_{\text{OPT}} \leq M$).

$G = (V, E)$ Baum, (s_i, t_i) , $i = 1, \dots, k$, $c: E \rightarrow \mathbb{Q}_0^+$ Kapazitäten. Suchen Multiflow (= Mehrgüterfluss) mit maximalem Wert.

Wenn primale und duale Schlupfbedingungen für einen Multiflow \mathcal{F} und einen Multicut \mathcal{M} gelten, so gilt $F = M$ ($F = \text{Wert von } \mathcal{F}$, $M = \text{Wert von } \mathcal{M}$), sowie $M_{\text{OPT}} = M$, $F_{\text{OPT}} = F$.

Proposition 15.2. *Gelte*

- a) $\sum_{i=1}^k f_i q_i(e) = c(e)$ für alle $e \in E$ mit $d_e > 0$ und
- b) $1 \leq \sum_{e \in E} d_e q_i(e) \leq 2$ für alle $i = 1, \dots, k$ mit $f_i > 0$.

Dann gilt für die zugehörigen Fluss- bzw. Multicutwerte F und M :

$$F \geq \frac{1}{2} \cdot F_{\text{OPT}} \text{ und } M \leq 2 \cdot M_{\text{OPT}}.$$

Beweis. Es gilt

$$\begin{aligned} F_{\text{OPT}} &\leq F^* \stackrel{\text{Dualitätssatz}}{=} M^* \leq M_{\text{OPT}} \leq M = \sum_{e \in E} d_e c(e) \\ &\stackrel{\text{a)}}{=} \sum_{e \in E} d_e \left(\sum_{i=1}^k f_i q_i(e) \right) \\ &= \sum_{i=1}^k f_i \underbrace{\left(\sum_{e \in E} d_e q_i(e) \right)}_{\leq 2 \text{ wegen b)}} \\ &\leq 2 \sum_{i=1}^k f_i = 2F \leq 2F_{\text{OPT}} \leq 2M_{\text{OPT}}. \end{aligned}$$

Also

$$F \geq \frac{1}{2} \cdot F_{\text{OPT}} \text{ und } M \leq 2 \cdot M_{\text{OPT}}.$$

□

Algorithmus Flowcut (von Garg, Yannakakis und Vazirani)

1. Wähle $r \in V$ als Wurzel.

2. **Phase 1:** Flusskonstruktion

Wir gehen von den Blättern von G zur Wurzel r und routen dabei soviel Fluss wie möglich.

Iteration: Für die Level $l = l_{\max}, l_{\max} - 1, \dots, 0$.

Sei $v \in V$ ein Knoten im aktuellen Level l . Für jedes Quellen/Senken-Paar (s_i, t_i) im Teilbaum T_v (mit Wurzel v) schicke über den Pfad P_i soviel Fluß wie möglich, dabei ist die Reihenfolge, in der verschiedene Quellen/Senken-Paar in T_v bedient werden beliebig.

Sei \mathcal{F} der so erzeugte Fluss mit Teilflusswerten f_i und Gesamtflusswert $F = \sum_{i=1}^k f_i$.

3. Sei Q_v die Menge der ein Teilbaum T_v saturierten Kanten. Sei I_v die Menge der Quellen/Senken-Paare in T_v .

4. **Phase 2:** Multicutkonstruktion.

In dieser Phase gehen wir von der Wurzel zu den Blättern, levelweise und wählen saturierte Kante für den Multicut wie folgt:

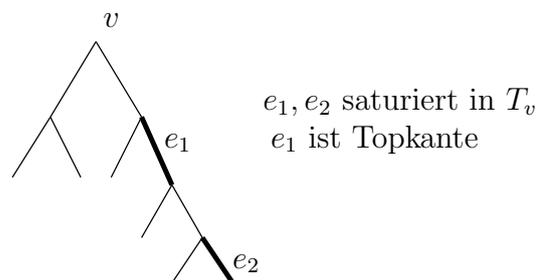
Iteration: Für $l = 0, \dots, l_{\max}$ führe aus:

Ist l das aktuelle Level und $v \in V$ mit $l(v) = l$, so wähle $e \in LIM(v)$ (siehe Definition unten) genau dann, wenn *keine* Kante auf dem Pfad von e nach v schon gewählt wurde. Sei \mathcal{M} die Menge der gewählten Kanten und M sei die Kapazität von \mathcal{M} .

Definition. Eine Kante $e \in Q_v$ heisst *Topkante*, wenn e keine Vorgängerkante in Q_v besitzt.

Wir definieren $LIM(v)$ als die Kardinalitätsmaximale Menge von Topkanten aus Q_v und nennen sie *Grenzkantenmenge*.

Beispiel.



Lemma 15.3. Sei $\mathcal{M}' = \bigcup_{v \in V} LIM(v)$.

- (i) \mathcal{M}' und \mathcal{M} (das vom Algorithmus gelieferte) bilden jeweils einen Multicut.
- (ii) Sei P ein s_i - t_i -Flusspfad und T_v der Teilbaum mit größtem Level $l(v)$, der s_i und t_i enthält. Sei $u \in V$ mit $LIM(u) \cap E(P) \neq \emptyset$ ($E(P)$ ist die Menge der Kanten in P). Dann ist u ein Vorgängerknoten von v .
- (iii) Ist $u \in V$ ein Vorgängerknoten von $v \in V$, dann ist keine Kante aus $LIM(v)$ Vorgängerkante einer Kante aus $LIM(u)$.

Beweis. (i) $\mathcal{M}'' := \bigcup_{v \in V} Q_v$ ist ein Multicut. Da \mathcal{M}' Topkanten enthält, ist auch \mathcal{M}'' ein Multicut.

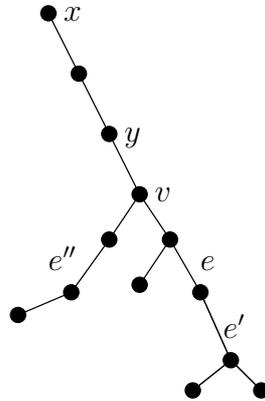
Annahme: \mathcal{M} ist kein Multicut. Dann gibt es s_i und t_i , so dass P_i \mathcal{M} nicht passiert. Da \mathcal{M}'' Multicut ist, gibt es ein $v \in V$ mit $E(P_i) \cap LIM(v) \neq \emptyset$, d.h. es gibt eine Kante $e \in LIM(v)$, die saturiert ist und auf P_i liegt. Wegen der Annahme gilt, $e \notin \mathcal{M}$. Es folgt, dass es eine saturierte Vorgängerkante $e' \in E(P_i)$ gibt, die in Phase 2 zu \mathcal{M} hinzugenommen wurde. Widerspruch zur Annahme.

- (ii) Annahme: u ist kein Vorgängerknoten von v . Es folgt, dass u in Phase 1 vor v betrachtet wurde. Ausserdem wurde $e \in LIM \cap E(P)$ durch irgendeinen s_j - t_j -Fluss, $j \neq i$, saturiert (in T_n). Widerspruch zu P s_i - t_i -Flusspfad.
- (iii) Annahme: Es gibt eine Kante $e \in LIM(v)$, die Vorgängerkante einer Kante $e' \in LIM(u)$ ist. Zum Zeitpunkt, wenn T_v betrachtet wird, wird e saturiert. Da $e' \in LIM(u)$ und u Vorgänger von v , wird e' später als e saturiert. Widerspruch, da e' nicht später als e saturiert werden kann. \square

Lemma 15.4. Für (s_i, t_i) sei T_v der Teilbaum, der s_i und t_i enthält und maximales Level $l(v)$ besitzt. Dann gilt, \mathcal{M} enthält höchstens zwei Kanten von P_i für jedes $i = 1, \dots, k$.

Beweis. Annahme: Es existiert ein i , so dass \mathcal{M} drei Kanten von P_i enthält (o.B.d.A.). e, e' und e'' seien diese Kanten. Dann gibt es $x, y \in V$, so dass $e \in LIM(x)$ und $e' \in LIM(y)$. Mit Lemma 15.3 (iii) folgt, dass x ein Vorgängerknoten von y ist. Ferner folgt mit Lemma 15.3 (ii), dass x und y Vorgängerknoten von v sind. Widerspruch dazu, dass e' gewählt wurde,

obwohl e schon gewählt wurde.



□

Satz 15.5 (Garg, Vazirani, Yannakakis 1994). *Flowcut terminiert in Zeit $\mathcal{O}(n)$ und es gilt $F \geq \frac{1}{2} \cdot F_{\text{OPT}}$ und $M \leq 2 \cdot M_{\text{OPT}}$.*

Beweis. Lemma 15.4 und Proposition 15.2.

□

16 Packen und Überdecken in Hypergraphen

16.1 Chernoff-Hoeffding-Ungleichungen

Es seien X_1, \dots, X_n unabhängige Zufallsvariablen mit $a_k \leq X_k \leq b_k$, $a_k, b_k \in \mathbb{R}$ für alle k . Weiterhin sei $X = \sum_{i=1}^n X_i$ und $\mathbb{E}(X)$ der Erwartungswert von X .

Satz 16.1 (Hoeffding, 1964). *Für jedes $t > 0$ gilt,*

$$(i) \Pr[X - \mathbb{E}(X) \geq t] \leq e^{-\frac{2t^2}{c^2}} \text{ und}$$

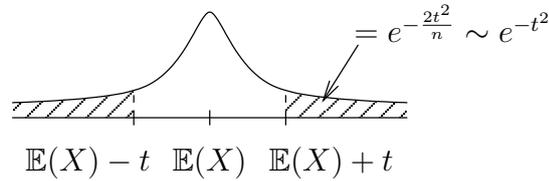
$$(ii) \Pr[X - \mathbb{E}(X) \leq -t] \leq e^{-\frac{2t^2}{c^2}},$$

wobei $c^2 = \sum_{k=1}^n (b_k - a_k)^2$.

Korollar 16.2 (Chernoff-Ungleichung). *Für $a_k = 0, b_k = 1$ für alle k gilt für jedes $t > 0$,*

$$(i) \Pr[X - \mathbb{E}(X) \geq t] \leq e^{-\frac{2t^2}{n}} \text{ und}$$

$$(ii) \Pr[X - \mathbb{E}(X) \leq -t] \leq e^{-\frac{2t^2}{n}}.$$



Falls man für die Large-Deviation in t exponentielles Abfallen hat, so sagt man, die Zufallsvariable X ist *scharf um den Erwartungswert konzentriert*. Man hat hier jeweils von $\mathbb{E}(X) \pm t$ eine Verteilung, die eine Dichte der Form $\sim e^{-t^2}$, also ähnlich der Dichte der Gaußverteilung besitzt.

Konsequenz: Die Zufallsvariable X ist “gut” mit dem Erwartungswert “identifizierbar”.

Vergleich mit Markow-Ungleichung¹:

$$\Pr[X \geq \mathbb{E}(X) + t] \leq \frac{\mathbb{E}(X)}{\mathbb{E}(X) + 1} = \frac{1}{1 + \frac{t}{\mathbb{E}(X)}} = \frac{1}{1 + \frac{t}{np}} \xrightarrow{t \rightarrow \infty} 0,$$

für $p := \mathbb{E}(X_k)$, $X_k \in \{0, 1\}$.

Satz 16.3 (Angluin-Valiant-Ungleichung). Seien X_1, \dots, X_n unabhängige Zufallsvariablen mit $0 \leq X_k \leq 1$ und $\mathbb{E}(X_k) = p_k$, $0 \leq p_k \leq 1$, für alle k . Sei $\beta \in (0, 1]$. Dann gilt

$$(i) \Pr[X \geq (1 + \beta)\mathbb{E}(X)] \leq e^{-\frac{\beta^2 \mathbb{E}(X)}{3}} \text{ und}$$

$$(ii) \Pr[X \leq (1 - \beta)\mathbb{E}(X)] \leq e^{-\frac{\beta^2 \mathbb{E}(X)}{2}}.$$

Der Vorteil von Satz 16.3 gegenüber Satz 16.2 ist eine bessere obere Schranke für $\mathbb{E}(X) \ll n$.

Mit Chernoff: $t = \beta \mathbb{E}(X)$, also $\Pr[X \geq (1 + \beta)\mathbb{E}(X)] \leq e^{-\frac{2\beta^2 \mathbb{E}(X)^2}{n}}$.

$$\frac{2\mathbb{E}(X)^2}{n} < \frac{\mathbb{E}(X)}{3} \Leftrightarrow \frac{6}{n} < \frac{1}{\mathbb{E}(X)} \Leftrightarrow \mathbb{E}(X) < \frac{n}{6}.$$

Die Güte der Hoeffding-Ungleichung (Satz 16.1) hängt von der Größe von $c^2 = \sum_{k=1}^n (b_k - a_k)^2$, d.h. von den $|b_k - a_k|$ ab.

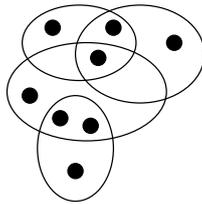
¹Markow-Ungleichung: $\Pr[X \geq \lambda] \leq \frac{\mathbb{E}(X)}{\lambda}$

16.2 Matching in Hypergraphen

Sei $\mathcal{H} = (V, \mathcal{E})$ ein Hypergraph mit $|V| = n$, $|\mathcal{E}| = m$. Eine Teilmenge der Kantenmenge $\mathcal{E}' \subseteq \mathcal{E}$ heisst k -Matching in \mathcal{H} , wenn in jedem $v \in V$ höchstens k Kanten aus \mathcal{E}' inzident sind. Für $k = 1$ spricht man von einem *Matching*.

Falls \mathcal{H} ein Graph ist, hat man den bekannten Matchingbegriff aus der Graphentheorie.

Beispiel.



Problem (Maximum- k -Matching). Problem: Gegeben eine Gewichtsfunktion $w: E \rightarrow [0, 1] \cap \mathbb{Q}$. Finde ein k -Matching $M \subseteq \mathcal{E}$ in \mathcal{H} mit maximalem Gewicht $w(M)$.

Die Entscheidungsversion des Problems ist NP-hart, schon für den Fall, dass $|E| = 3$ für alle $E \in \mathcal{E}$ und $k = 1$.

Problem (ILP). Sei $A = (a_{i,j})_{i \in \mathcal{E}, j \in V}$ die Inzidenzmatrix von \mathcal{H} , d.h.

$$a_{i,j} = \begin{cases} 1 & j \in i \\ 0 & \text{sonst} \end{cases}$$

für alle i, j .

$$\begin{aligned} \max \quad & \sum_{i=1}^m w_i X_i \\ \sum_{i=1}^m a_{i,j} X_i & \leq k \quad \forall j \in V \\ X_i & \in \{0, 1\} \quad \forall i \in \mathcal{E} \end{aligned}$$

Beziehung: (ILP) \Leftrightarrow Maximum- k -Matching.

$\mathcal{E}' := \{i \in \mathcal{E} \mid X_i = 1\}$, dann $\sum w_i X_i = w(\mathcal{E}')$.

(LP) ist die Relaxierung von (ILP) mit $X_i \in [0, 1] \cap \mathbb{Q}$ für alle i .

(LP) ist (z.B. mit Ellipsoidalgorithmus) optimal lösbar. Sei $y = (y_1, \dots, y_m)$ eine optimale Lösung von (LP). Sei $\text{OPT}^* = \sum w_i y_i$, sowie OPT der Wert einer optimalen Lösung von (ILP). Dann gilt, $\text{OPT}^* \geq \text{OPT}$.

$y = (y_1, \dots, y_m)$, $0 \leq y_i \leq 1$ für alle i .

Rundungsmöglichkeiten:

a) Kaufmännisches (naives) Runden:

$$y_i \geq \frac{1}{2} \Rightarrow x_i = 1,$$

$$y_i < \frac{1}{2} \Rightarrow x_i = 0.$$

Damit erhalten wir $M = \{i \mid X_i\} \subseteq \mathcal{E}$. Leider ist M im Allgemeinen *kein* k -Matching und $w(M)$ kann beliebig weit von OPT entfernt sein.

b) Randomisiertes Runden:

$$x_i = 1 \text{ mit W'keit } y_i,$$

$$x_i = 0 \text{ mit W'keit } (1 - y_i)$$

für alle i unabhängig.

Algorithmus Randomisiertes Runden

Wähle $\epsilon \in (0, 1)$.

Für $i = 1, \dots, m$ führe unabhängig aus:

$$X_i = 1 \text{ mit W'keit } (1 - \frac{\epsilon}{2})y_i$$

$$X_i = 0 \text{ mit W'keit } 1 - (1 - \frac{\epsilon}{2})y_i$$

Lemma 16.4. (i) $\mathbb{E}(\sum_{i=1}^m w_i X_i) = (1 - \frac{\epsilon}{2})\text{OPT}^* \geq (1 - \frac{\epsilon}{2})\text{OPT}$.

(ii) $\mathbb{E}(\sum_{i=1}^m a_{i,j} X_i) \leq (1 - \frac{\epsilon}{2})k$ für alle $j \in V$.

Beweis. (i)

$$\begin{aligned} \mathbb{E}(\sum_{i=1}^m w_i X_i) &= \sum_{i=1}^m w_i \mathbb{E}(X_i) = \sum_{i=1}^m w_i (1 - \frac{\epsilon}{2})y_i \\ &= (1 - \frac{\epsilon}{2}) \sum_{i=1}^m w_i y_i = (1 - \frac{\epsilon}{2})\text{OPT}^* \end{aligned}$$

(ii)

$$\begin{aligned}\mathbb{E}\left(\sum_{i=1}^m a_{i,j} X_i\right) &= \sum_{i=1}^m a_{i,j} \mathbb{E}(X_i) = \sum_{i=1}^m a_{i,j} \left(1 - \frac{\epsilon}{2}\right) y_i \\ &= \left(1 - \frac{\epsilon}{2}\right) \sum_{i=1}^m a_{i,j} y_i < \left(1 - \frac{\epsilon}{2}\right) k,\end{aligned}$$

da y Lösung des LP ist. □

Frage: Mit welcher W'keit gilt: $\sum w_i X_i \geq \alpha \text{OPT}$ und $\sum a_{i,j} X_i \leq k$ für alle j ?

Satz 16.5. Sei $k \geq \frac{12}{\epsilon^2} \ln(4n)$. Der Algorithmus Randomisiertes Runden berechnet in Polynomieller Zeit ein k -Matching M in \mathcal{H} , so dass mit Wahrscheinlichkeit mindestens $\frac{1}{2}$ gilt, $w(M) \geq (1 - \epsilon)\text{OPT}$.

Beweis. Polynomielle Laufzeit ist klar.

Es gilt dann $M = \{i \in \mathcal{E} \mid X_i = 1\}$. Wir zeigen nun

a) $\Pr[\exists j \in V : \sum_{i=1}^m a_{i,j} X_i > k] \leq \frac{1}{4}$.

b) $\Pr[\sum_{i=1}^m w_i X_i < (1 - \epsilon)\text{OPT}] \leq \frac{1}{4}$.

Daraus folgt dann, dass

$$\Pr\left[\sum_{i=1}^m w_i X_i \geq (1 - \epsilon)\text{OPT} \text{ und } \sum_{i=1}^m a_{i,j} X_i \leq k \forall j\right] \geq 1 - \left(\frac{1}{4} + \frac{1}{4}\right) = \frac{1}{2}.$$

zu a) Sei $j \in V$ beliebig.

$$\Pr\left[\sum_{i=1}^m a_{i,j} X_i > k\right] \stackrel{\beta := \frac{\epsilon}{2-\epsilon}}{=} \Pr\left[\sum_{i=1}^m a_{i,j} X_i > (1 + \beta)\left(1 - \frac{\epsilon}{2}\right)k\right].$$

Aus Lemma 16.4 folgt, $\mathbb{E}(\sum a_{i,j} X_i) \leq \left(1 - \frac{\epsilon}{2}\right)k$. Also

$$\Pr\left[\sum_{i=1}^m a_{i,j} X_i > (1 + \beta)\left(1 - \frac{\epsilon}{2}\right)k\right] \leq \Pr\left[\sum_{i=1}^m a_{i,j} X_i > (1 + \beta)\mathbb{E}\left(\sum_{i=1}^m a_{i,j} X_i\right)\right].$$

O.B.d.A. gilt $\mathbb{E}(\sum a_{i,j} X_i) = \left(1 - \frac{\epsilon}{2}\right)k$. Dann gilt

$$\Pr\left[\sum_{i=1}^m a_{i,j} X_i > (1 + \beta)\mathbb{E}\left(\sum_{i=1}^m a_{i,j} X_i\right)\right] \leq e^{-\frac{\beta^2 \mathbb{E}(\sum a_{i,j} X_i)}{3}} = e^{-\frac{\beta^2 (1 - \frac{\epsilon}{2})k}{3}}$$

$$\stackrel{\text{Anghwin-Valiant}}{\leq} \frac{1}{4n}.$$

$$\begin{aligned} \Pr[\exists j : \sum a_{i,j} X_i > k] &= \Pr[\bigcup_j : \sum a_{i,j} X_i > k] \\ &\leq \sum_j \underbrace{\Pr[\sum a_{i,j} X_i > k]}_{\leq \frac{1}{4n}} \leq \frac{1}{4n} n = \frac{1}{4}. \end{aligned}$$

zu b) Übung. □

Offene Probleme

1. $k \ll \mathcal{O}(\frac{\ln n}{\epsilon^2})$?
2. Gibt es einen randomisierten Algorithmus, der ein perfektes Matching in einem Hypergraphen *minimalen* Gewichtes findet? Für Graphen: Ja!

Wir beweisen die Hoeffding-Ungleichung.

$$\begin{aligned} \Pr\left[\sum_{k=1}^n X_k > \mu + \lambda\right] &\leq e^{-\frac{2\lambda^2}{\sum_{k=1}^n (b_k - c_k)^2}}, \\ \Pr\left[\sum_{k=1}^n X_k < \mu - \lambda\right] &\leq e^{-\frac{2\lambda^2}{\sum_{k=1}^n (b_k - c_k)^2}}, \end{aligned}$$

für $\mu = \mathbb{E}(\sum X_k)$. (X_1, \dots, X_n unabhängige Zufallsvariablen mit $a_k \leq X_k \leq b_k$).

Lemma 16.6. Sei X eine Zufallsvariable mit $a \leq X \leq b$ und $\mathbb{E}(X) = 0$. Dann gilt für jedes $h > 0$:

$$\mathbb{E}(e^{hX}) \leq e^{\frac{1}{8}h^2(b-a)^2}.$$

e^{hX} heisst *Moment-erzeugende Funktion* von X .

$$e^{hX} = 1 + h \underbrace{X}_{\text{1. Moment}} + \frac{h^2}{2} \underbrace{X^2}_{\text{2. Moment}} + \dots + \frac{h^l}{l!} \underbrace{X^l}_{\text{l-tes Moment}} + \dots$$

$$\begin{aligned} \mathbb{E}(e^{hX}) &= 1 + h\mathbb{E}(X) + \frac{h^2}{2!}\mathbb{E}(X^2) + \dots \\ &\sim e^{h\mathbb{E}(X)} \end{aligned}$$

Beweis.

$$X = \frac{X-a}{b-a}b + \frac{b-X}{b-a}a.$$

Da \exp konvex ist gilt (für konvexes f gilt: $f(\alpha a + (1-\alpha)b) \leq \alpha f(a) + (1-\alpha)f(b)$)

$$e^{hX} \leq \frac{X-a}{b-a}e^{hb} + \frac{b-X}{b-a}e^{ha}$$

Also

$$\mathbb{E}(e^{hX}) \leq -\frac{a}{b-a}e^{hb} + \frac{b}{b-a}e^{ha} = e^{f(y)}$$

für $f(y) = -py + \ln(1-p+pe^y)$, $p = -\frac{a}{b-a}$, $y = (b-a)h$. Taylorentwicklung von f :

$$f(y) = f(0) + \frac{f'(0)}{1}y + \frac{f''(\zeta)}{2!}y^2$$

für gewisses $\zeta \in [0, y]$. Wir bilden die Ableitung von f :

$$f'(x) = -p + \frac{pe^x}{1-p+pe^x}.$$

Es gilt

$$f(0) = f'(0) = 0.$$

Die zweite Ableitung von f ist:

$$f''(x) = \frac{p(1-p)e^x}{(p+(1-p)e^x)^2}.$$

Wenn $f''(\zeta) \leq \frac{1}{4}$, dann folgt die Behauptung.

Wir wollen haben, dass

$$\begin{aligned} \frac{\kappa \cdot \nu}{(\kappa + \nu)^2} &\leq \frac{1}{4} \\ \Leftrightarrow \kappa \nu &\leq \frac{1}{4}(\kappa + \nu)^2 \\ \Leftrightarrow \sqrt{\kappa \nu} &\leq \frac{\kappa + \nu}{2}. \end{aligned}$$

Dies gilt, da das geometrische Mittel höchstens so groß ist wie das arithmetische Mittel. Für $\kappa = p$ und $\nu = (1-p)e^x$ folgt dann, dass $f''(\zeta) \leq \frac{1}{4}$ ist. \square

Beweis der Hoeffding-Ungleichung. $S_n := \sum_{k=1}^n X_k$.

$$\Pr\left[\sum X_k \geq \mu + \lambda\right] = \Pr[S_n \geq \mu + \lambda] \stackrel{\text{Markow-Ungl.}}{\leq} e^{-h\lambda} \mathbb{E}(e^{h(S_n - \mu)}).$$

Behauptung:

$$\mathbb{E}(e^{h(S_n - \mu)}) \leq e^{h^2 \sum_{k=1}^n (b_k - a_k)^2}.$$

Falls die Behauptung gilt, haben wir

$$\Pr\left[\sum X_k \geq \mu + \lambda\right] \leq e^{-h\lambda + h^2 \sum_k (b_k - a_k)^2} \stackrel{\text{geeignetes } h}{\leq} e^{\frac{-2\lambda^2}{\sum (b_k - a_k)^2}}.$$

Zur Behauptung:

$$\begin{aligned} \mathbb{E}(e^{h(S_n - \mu)}) &= \mathbb{E}(e^{h[X_1 - \mathbb{E}(X_1)] + (X_2 - \mathbb{E}(X_2)) + \dots + (X_n - \mathbb{E}(X_n))}) \\ &= \mathbb{E}\left(\prod_{k=1}^n e^{h(X_k - \mathbb{E}(X_k))}\right) \\ &\stackrel{X_i \text{ unabh.}}{=} \prod_{k=1}^n \mathbb{E}(e^{h(X_k - \mathbb{E}(X_k))}) \\ &\stackrel{\text{Lemma 16.5}}{\leq} e^{\frac{1}{8} h^2 \sum_{k=1}^n (b_k - a_k)^2}. \end{aligned}$$

□

17 Überdeckungen in Hypergraphen

Sei $\mathcal{H} = (V, \mathcal{E})$ ein Hypergraph mit $n = |V|$, $m = |\mathcal{E}|$. Sei $w: V \rightarrow \mathbb{Q}_0^+$ eine Kostenfunktion. Fasse w auch als Vektor aus $(\mathbb{Q}_0^+)^n$ auf.

Definition 17.1. Ein *Vertex-Cover* (Hitting-Set) in \mathcal{H} ist eine Menge $X \subseteq V$, so dass jede Hyperkante aus \mathcal{E} in X inzidiert. Die *Kosten* (oder Gewicht) $w(X)$ sind die Summe $\sum_{v \in X} w(v)$.

Problem 17.2 (Vertex-Cover in Hypergraphen). Eingabe: $\mathcal{H} = (V, \mathcal{E})$, w

Gesucht: Ein Vertex-Cover von \mathcal{H} mit minimalem Gewicht, das mit $\tau(\mathcal{H})$ bezeichnet wird.

Aus Kombinatorischer Optimierung I wissen wir, dass wenn \mathcal{H} ein bipartiter Graph ist, dann ist das Problem in \mathbf{P} , und für den ungewichteten Fall ($w \equiv 1$) gilt $\nu(\mathcal{H}) = \tau(\mathcal{H})$.

Für $w \equiv 1$ heisst $\tau(\mathcal{H})$ die *Überdeckungsanzahl* (oder auch blocking oder covering number) von \mathcal{H} .

In allgemeinen Graphen ist das Vertex-Cover-Problem **NP**-schwer, also erst recht für Hypergraphen. Für Graphen gibt es allerdings einen einfachen 2-Faktor Approximationsalgorithmus.

Lund und Yannakakis haben 1996 ein nicht Approximationsresultat bewiesen: Für jedes $\alpha < \frac{1}{4}$ gibt es keinen polynomiellen ($\alpha \log m$)-Approximationsalgorithmus für das Vertex-Cover-Problem in Hypergraphen, es sei denn $\text{NP} \subseteq \text{DTIME}(n^{\text{polylog } n})$.

Problem (VC-ILP). Sei $A = (a_{i,j}) \in \{0, 1\}^{m \times n}$ die Kanten-Knoten-Inzidenzmatrix von \mathcal{H} .

$$\begin{aligned} \min \sum_{j=1}^n w_j x_j \\ \sum_{j=1}^n a_{i,j} x_j \geq 1 \quad \forall i \in \{1, \dots, m\} \\ x_j \in \{0, 1\} \quad \forall j \in \{1, \dots, n\} \end{aligned}$$

Damit haben wir, dass (VC) äquivalent ist zu (VC-ILP).

Algorithmus (RR-Vertex-Cover). 1. Löse die LP-Relaxierung von VC-ILP. Sei $y \in [0, 1]^n$ eine optimale Lösung mit Wert $\text{OPT}^* = w^T y$.

2. Erzeuge $X_1, \dots, X_n \in \{0, 1\}$ durch randomisiertes Runden:
Wähle $k \in \mathbb{N}$ fest.

$$\begin{aligned} \Pr[X_j = 1] &= 1 - (1 - y_j)^k \\ \Pr[X_j = 0] &= (1 - y_j)^k \end{aligned}$$

für alle X_1, \dots, X_n unabhängig.

3. Ausgabe ist der Vektor $x = (X_1, \dots, X_n)$.

Für $k = 1$ erhalten wir das übliche Randomisierte-Runden.

Für $k = 2$ und z.B. für $y_j = 0.7$ erhalten wir

$$\Pr[X_j = 1] = 1 - (1 - 0.7)^2 = (1 - 0.3)^2 = 1 - \frac{1}{9} = \frac{8}{9}.$$

Im Vergleich zu $k = 1$

$$\Pr[X_j = 1] = 0.7 = \frac{7}{10}.$$

k beschreibt die k -fache Wiederholung des zufälligen Experimentes, $(1 - y_j)^k$ ist die Wahrscheinlichkeit, dass in k Würfeln stets $X_j = 0$ ist, und $1 - (1 - y_j)^k$ ist die Wahrscheinlichkeit, dass mindestens *einer* der k Würfe eine Eins bringt.

Frage: Ist x ein Vertex-Cover und wie groß ist $w^T x$ im Vergleich zu OPT^* ?

Für $j \in V$ sei $D_j := \{E \in \mathcal{E} \mid j \in E\}$ und $D = \max_{1 \leq j \leq n} |D_j|$. ($|D_j|$ ist der Grad von j , auch notiert durch $\text{deg}(j)$).

Sei A_i das (schlechte) Ereignis, dass der die i -te Ungleichung von VC-ILP durch x verletzt wird, d.h. $\sum_{j=1}^n a_{i,j} x_j = 0$. Sei $B_i := A_i^c$ und $F = \bigcap_{i=1}^m B_i$. F ist das Ereignis, dass x ein Vertex-Cover ist, oder mit anderen Worten, x ist zulässig für VC-ILP.

Lemma 17.3.

$$\mathbb{E}(w^T x \mid F) = \sum_{j=1}^n w_j \frac{\Pr[F \mid X_j = 1]}{\Pr[F]} \cdot \Pr[X_j = 1].$$

Beweis. Es gilt

$$\mathbb{E}\left(\sum_j w_j X_j \mid F\right) = \sum_j w_j \mathbb{E}(X_j \mid F) = \sum_j w_j \Pr[X_j = 1 \mid F].$$

Mit

$$\Pr[A \mid B] = \frac{\Pr[A \cap B]}{\Pr[B]}$$

folgt die Behauptung. □

Lemma 17.4 (Korrelationsungleichung). Sei $I \subseteq \{1, \dots, m\}$, $I' \subseteq I$, $\tilde{I} = I \setminus I'$. Dann gilt

$$\Pr\left[\bigcap_{i \in I} B_i\right] \geq \Pr\left[\bigcap_{i \in I'} B_i\right] \cdot \Pr\left[\bigcap_{i \in \tilde{I}} B_i\right].$$

Beweis. In der Übung. Erläuterung für $m = 2$: Dann soll gelten, dass

$$\Pr[B_1 \cap B_2] \geq \Pr[B_1] \cdot \Pr[B_2]. \quad (*)$$

Für unabhängige B_1 und B_2 gilt hier sogar Gleichheit. □

Die Ungleichung (*) gilt genau dann, wenn

$$\Pr[B_1] \leq \frac{\Pr[B_1 \cap B_2]}{\Pr[B_2]} = \Pr[B_1 \mid B_2].$$

D.h. die Wahrscheinlichkeit, dass ein Ereignis (hier B_1) eintritt, wird durch das Erfülltsein eines anderen Ereignisses (hier B_2) hochgetrieben.

Lemma 17.5. (i) $(1 - a)^b \geq 1 - ab$ für alle $a \in [0, 1]$, $b \in \mathbb{R}$, $b \geq 1$.

(ii) $1 + a \leq e^a$ für alle $a \in \mathbb{R}$, sogar strikte Ungleichung für $a \neq 0$.

(iii) $\Pr[B_i] \geq 1 - e^{-k}$ für alle $i \in \{1, \dots, m\}$.

Beweis. (i) siehe Analysis I

(ii) siehe Analysis I

(iii)

$$\begin{aligned} \Pr[A_i] &= \Pr\left[\sum_{j=1}^n a_{i,j} X_j = 0\right] \stackrel{(1)}{=} \Pr\left[\bigcap_{j \in E_i} [X_j = 0]\right] = \prod_{j \in E_i} \Pr[X_j = 0] \\ &= \prod_{j \in E_i} (1 - y_j)^k \stackrel{(ii)}{\leq} \prod_{j \in E_i} e^{-ky_j} = e^{-k \sum_{j \in E_i} y_j} = e^{-k \sum_{j=1}^n a_{i,j} y_j} \\ &\stackrel{(2)}{\leq} e^{-k} \end{aligned}$$

zu (1): E_i sei die i -te Hyperkante.

zu (2): $\sum_{j=1}^n a_{i,j} y_j \geq 1$, wegen y LP-Lösung.

Es folgt $\Pr[B_i] = 1 - \Pr[A_i] \geq 1 - e^{-k}$. □

$D_j = \{E \in \mathcal{E} \mid j \in E\}$, $\deg j = |D_j|$. $D := \max_{1 \leq j \leq n} \deg j$.

Satz 17.6 (Bertsimas, Vokrei '95). Für $k := \ln D$ gilt

$$\mathbb{E}[w^T x \mid F] \leq \frac{\ln D}{\left(1 - \frac{1}{D}\right)^D} \text{OPT}^*.$$

Beweis. Sei $j \in \{1, \dots, n\}$. Aus $X_j = 1$ folgt, dass das Ereignis $\bigcap_{i \in D_j} B_i$

erfüllt ist. Es folgt,

$$\begin{aligned}
\Pr[F \mid X_j = 1] &= \Pr\left[\bigcap_{i=1}^m B_i \mid X_j = 1\right] \\
&= \Pr\left[\bigcap_{i \in \{1, \dots, m\} \setminus D_j} B_i \cap \bigcap_{i \in D_j} B_i \mid X_j = 1\right] \\
&= \frac{\Pr\left[\bigcap_{i \in \{1, \dots, m\} \setminus D_j} B_i \cap \bigcap_{i \in D_j} B_i \cap [X_j = 1]\right]}{\Pr[X_j = 1]}
\end{aligned}$$

(Für $i \in \{1, \dots, m\} \setminus D_j$ sind B_i und $[X_j = 1]$ unabhängig.)

$$= \Pr\left[\bigcap_{i \in \{1, \dots, m\} \setminus D_j} B_i\right].$$

$$\begin{aligned}
\mathbb{E}[w^T x \mid F] &\stackrel{\text{Lemma 17.3}}{=} \sum_{j=1}^n w_j \frac{\Pr[F \mid X_j = 1]}{\Pr[F]} \Pr[X_j = 1] \\
&= \sum_{j=1}^n w_j \frac{\Pr\left[\bigcap_{i \in \{1, \dots, m\} \setminus D_j} B_i\right] \Pr[X_j = 1]}{\Pr\left[\bigcap_{i \in \{1, \dots, m\}} B_i\right]}.
\end{aligned}$$

$$\begin{aligned}
\Pr\left[\bigcap_{i \in \{1, \dots, m\}} B_i\right] &= \Pr\left[\bigcap_{i \in \{1, \dots, m\} \setminus D_j} B_i \cap \bigcap_{i \in D_j} B_i\right] \\
&\stackrel{\text{Lemma 17.4}}{\geq} \Pr\left[\bigcap_{i \in \{1, \dots, m\} \setminus D_j} B_i\right] \Pr\left[\bigcap_{i \in D_j} B_i\right].
\end{aligned}$$

Also

$$\begin{aligned} \mathbb{E}[w^T x \mid F] &\stackrel{\text{Lemma 17.4}}{\leq} \sum_{j=1}^n w_j \frac{\Pr[X_j = 1]}{\prod_{i \in D_j} \Pr[B_i]} \\ &\leq \sum_{j=1}^n w_j \frac{\Pr[X_j = 1]}{(1 - e^{-k})^{|D_j|}} \leq \sum_{j=1}^n w_j \frac{\Pr[X_j = 1]}{(1 - e^{-k})^D} \end{aligned}$$

(Lemma 17.5 sagt: $\Pr[B_i] \geq 1 - e^{-k}$ für alle i .)

$$\begin{aligned} &= \sum_{j=1}^n w_j \frac{1 - (1 - y_j)^k}{(1 - e^{-k})^D} \stackrel{\text{Lemma 17.5}}{\leq} \sum_{j=1}^n w_j \frac{(1 - (1 - ky_j))}{(1 - e^{-k})^D} \\ &= k \sum_{j=1}^n w_j y_j (1 - e^{-k})^{-D} = k(1 - e^{-k})^{-D} \text{OPT}^* \\ &= \ln(D) \left(1 - \frac{1}{D}\right)^{-D} \text{OPT}^*. \end{aligned}$$

□

Man kann mit Hpefding erreichen, dass mit W 'keit $\geq p > 0$ gilt: Lösung ist zulässig und $w^T x \leq c' \ln(D) \text{OPT}^*$.

1. Variation: Finde $V' \subseteq V$, so dass mindestens $\alpha \cdot m$ Hyperkanten in V' inzident sind, $\alpha > 0$ und Gewicht $w(V')$ minimal.
2. Variation: (ϵ -Netzproblem) Finde $V' \subseteq V$ mit minimalem Gewicht $w(V')$, so dass alle Hyperkanten $E \in \mathcal{E}$ mit $|E| \geq \epsilon|V|$ in V' inzident sind. ($0 < \epsilon < 1$)

Für ϵ -Netz V' ist die Frage: Wie gross ist $|V'|$ als Funktion von $n, \frac{1}{\epsilon}$? Insbesondere, kann man erreichen, dass $|V'|$ *nur* eine Funktion von $\frac{1}{\epsilon}$ ist? Wir hätten z.B. gerne, dass $|V'| = \mathcal{O}\left(\frac{1}{\epsilon} \ln \frac{1}{\epsilon}\right)$.

18 Max-Cut Problem

Problem (Max-Cut). Zu einem Graphen $G = (V, E)$ und einer Gewichtsfunktion $w: E \rightarrow \mathbb{Q}_0^+$ wird eine Partition V_1, V_2 von V gesucht, so dass

$$\sum_{\substack{x \in V_1, y \in V_2 \\ (x,y) \in E}} w((x,y)) \quad \text{maximal ist.}$$

Definition. Für $A \subseteq V$ sei

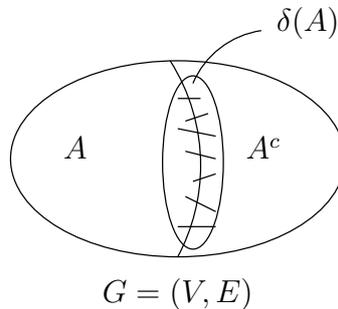
$$\delta(A) := \{(x, y) \in E \mid x \in A, y \in A^c\}$$

der durch A induzierte Schnitt, und sei

$$w(\delta(A)) := \sum_{e \in \delta(A)} w(e)$$

das Gewicht (oder auch Kapazität) des Schnittes $\delta(A)$.

Ungewichtete Version ($w \equiv 1$): Hier maximiert man $|\delta(A)|$. Anschaulich:



In Kombinatorische Optimierung I hatten wir gesehen, dass Max-Cut NP-schwer ist, aber es gibt einen Approximationsalgorithmus, der einen Schnitt $\delta(A)$ mit $w(\delta(A)) \geq \frac{1}{2} \text{OPT}$ in polynomieller Zeit findet. [$w(\delta(A)) \geq \frac{1}{2} |E|$]. Von Sahni und Gonzales '.

Eine offene Frage ist, ob man den Faktor $\frac{1}{2}$ verbessern kann. Diese Frage wurde 1994 durch Goemans und Williamson gelöst, durch Randomisierte Hyperebenen Runden und Semidefinite Relaxierung.

Grundidee: Sei $x_i \in \{-1, 1\}$, $i \in V$,

$$A := \{i \in V \mid x_i = 1\}, \quad A^c = \{i \in V \mid x_i = -1\}.$$

$$f(x_1, \dots, x_n) = \sum_{(i,j) \in E} w_{i,j} \frac{1 - x_i x_j}{2},$$

mit $w_{i,j} = w((i, j))$. Es gilt: $f(x_1, \dots, x_n) = w(\delta(A))$.

Problem (QP).

$$\begin{aligned} \max \quad & \sum_{(i,j) \in E} w_{i,j} \frac{1 - x_i x_j}{2} \\ \text{s.t.} \quad & x_i \in \{-1, 1\} \quad \forall i \in V \end{aligned}$$

Das quadratische Programm (QP) ist äquivalent zu Max-Cut.

Relaxiertes Programm:

Problem (RQP1).

$$\begin{aligned} \max \quad & \sum_{(i,j) \in E} w_{i,j} \frac{1 - x_i x_j}{2} \\ x_i \in & [-1, 1] \quad \forall i \in V \end{aligned}$$

Wir versuchen die Lösung des relaxierten Programms (RQP1) durch Ableitung zu finden:

Für den ungewichteten Fall ($w \equiv 1$) ist die Hessematrix von f :

$$(Hess(f))(x_0) = \frac{\partial f}{\partial x_i \partial x_j}(x_0) = a_{i,j} = \begin{cases} 1 & (i, j) \in E \\ 0 & \text{sonst} \end{cases}$$

D.h. $Hess(f)$ entspricht der Adjazenzmatrix A_G von G . Aber A_G ist im Allgemeinen indefinit. Damit gibt es möglicherweise viele lokale Extrema.

Also scheitert diese Herangehensweise. Der Grund wird im folgenden Satz klar:

Satz 18.1. *Es ist NP-schwer, (RQP1) zu lösen.*

Eine andere Relaxierung liefert folgendes Programm:

Problem (RQP2).

$$\begin{aligned} \max \quad & \sum_{(i,j) \in E} w_{i,j} \frac{1 - \vec{v}_i \cdot \vec{v}_j}{2} \\ \vec{v}_i \in & \mathbb{R}^n, \|\vec{v}_i\| = 1 \quad \forall i \in V \end{aligned}$$

Wobei n die Anzahl der Knoten ist, d.h. $n = |V|$. \cdot bezeichnet das Skalarprodukt.

(RQP2) ist eine Relaxierung von (QP), da die Lösungsmenge von (QP) eine Teilmenge der Lösungsmenge von (RQP2) ist.

Problem (RQP3).

$$\begin{aligned} \max \quad & \sum_{(i,j) \in E} w_{i,j} \frac{1 - y_{i,j}}{2} \\ Y = & (y_{i,j})_{1 \leq i, j \leq n} \\ y_{i,i} = & 1 \quad \forall i \in \{1, \dots, n\} \\ & Y \text{ positiv semidefinit} \end{aligned}$$

Wobei n die Anzahl der Knoten ist, d.h. $n = |V|$.

Proposition 18.2. (RQP3) ist äquivalent zu (RQP2).

Beweis. „ \Rightarrow “ Setze $Y := (y_{i,j})$ mit $y_{i,j} := \vec{v}_i \cdot \vec{v}_j$.
 Nun gilt $y_{i,i} = \vec{v}_i \cdot \vec{v}_i = \|\vec{v}_i\|^2 = 1$ für alle i .
 Y ist positiv semidefinit:

$$\forall x \in \mathbb{R}^n: \langle x, Yx \rangle = \sum_i \sum_j x_i x_j \vec{v}_i \cdot \vec{v}_j = \left\| \sum_i x_i \vec{v}_i \right\|^2 \geq 0.$$

„ \Leftarrow “ Sei Y positiv semidefinit mit $y_{i,i} = 1$ für alle i . Es gibt eine pos. semidef. Matrix B (auch $A^{\frac{1}{2}}$ genannt) mit $Y = B^T B$ (Choleskyzerlegung). Es gilt $y_{i,j} = (B^T B)_{i,j} = \vec{v}_i \cdot \vec{v}_j$ mit \vec{v}_i i -ter Zeilenvektor von B^T und \vec{v}_j j -ter Spaltenvektor von B . Nehme $\vec{v}_1, \dots, \vec{v}_n$ als die Zeilenvektoren von B^T . \square

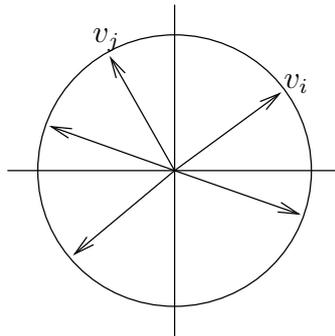
Satz 18.3. (RQP3) ist in polynomieller Zeit bis auf einen beliebigen relativen Fehler $\epsilon > 0$ für die Zielfunktion lösbar.

Beweisidee.

$$K := \{y = (y_{i,j}) \in \mathbb{R}^{n \times n} \mid y \text{ pos. def. und } y_{i,i} = 1 \forall i\}.$$

K ist eine konvexe Menge. Die Zielfunktion in (RQP3) ist linear über K . D.h. wir haben das Problem, eine lineare Zielfunktion über einer *konvexen* Menge zu maximieren. Mit einer Erweiterung des Ellipsoidalgorithmus lässt sich diese Aufgabe in Polynomialzeit lösen. \square

(RQP2) gibt uns also Vektoren $\vec{v}_1, \dots, \vec{v}_n \in S_n$ (S_n ist die n -dimensionale Sphäre).



$\vec{v}_i \cdot \vec{v}_j = \cos(\theta_{i,j})$, wobei $\theta_{i,j}$ der Winkel zwischen \vec{v}_i und \vec{v}_j ist, d.h. $\theta_{i,j} = \angle(\vec{v}_i, \vec{v}_j)$.

Algorithmus Random-Hyperplane

1. Wähle eine zufällige Hyperebene H durch den Nullpunkt in \mathbb{R}^n . Das wird dadurch realisiert, dass wir einen Normalenvektor \vec{r} zufällig unter Gleichverteilung wählen. (Das macht man so, dass die Komponenten von \vec{r} normalverteilt sind)

2.

$$V_1 = \{i \in V \mid \vec{v}_i \text{ liegt "unterhalb" von } H, \text{ d.h. } \langle \vec{v}_i, \vec{r} \rangle < 0\}.$$

$$V_2 = \{i \in V \mid \vec{v}_i \text{ liegt "oberhalb" von } H, \text{ d.h. } \langle \vec{v}_i, \vec{r} \rangle \geq 0\}.$$

Also $V_2 = V_1^c$.

3. Ausgabe: $\delta(V_1)$, $w(\delta(V_1))$.

Bemerkung: Es ist nicht trivial zu zeigen, dass \vec{r} gleichverteilt ist (siehe D. Knuth).

Lemma 18.4. *Es gilt*

$$\mathbb{E}[w(\delta(V_1))] = \sum_{(i,j) \in E} \frac{\arccos(\vec{v}_i \cdot \vec{v}_j)}{\pi} \cdot w_{i,j}.$$

Beweis.

$$\Pr[\text{sign} \langle \vec{v}_i, \vec{r} \rangle \neq \text{sign} \langle \vec{v}_j, \vec{r} \rangle] = 2 \cdot \frac{\theta}{2\pi} = \frac{1}{\pi} \arccos(\vec{v}_i \cdot \vec{v}_j),$$

wobei θ der Winkel zwischen \vec{v}_i und \vec{v}_j ist. □

Lemma 18.5. *Für $x \in [-1, 1]$ gilt:*

$$\frac{\arccos x}{\pi} \geq \alpha \cdot \frac{1-x}{2}$$

mit $\alpha \geq 0.87856$. Genau genommen gilt $\alpha = \min_{x \in [-1,1]} \frac{2 \arccos x}{\pi(1-x)}$.

Beweis. Sei $f(x) = \frac{2 \arccos x}{\pi(1-x)}$. Für $x_0 = -0.689\dots$ gilt $f'(x_0) = 0$ und $f''(x_0) > 0$. □

Satz 18.6. *Der Algorithmus Random-Hyperplane findet einen Schnitt $\delta(V_1)$ mit*

$$\begin{aligned} \mathbb{E}[w(\delta(V_1))] &\geq \alpha \cdot \text{OPT}(RQP2) \geq \alpha \cdot \text{OPT}(Max-Cut) \\ &\geq 0.87858 \cdot \text{OPT}(Max-Cut). \end{aligned}$$

Beweis. Aus Lemma 18.4 und 18.5 folgt, dass

$$\mathbb{E}[w(\delta(V_1))] \geq \alpha \cdot \sum_{(i,j) \in E} \frac{1 - \vec{v}_i \cdot \vec{v}_j}{2} \cdot w_{i,j} = \alpha \cdot \text{OPT}(\text{RQP2}).$$

□

Bemerkung. • Håstad 1996: Eine Approximation besser als 0.97... ist nicht möglich, es sei denn $P = NP$.

- Mahajan, Ramesh 1995: Algorithmus und Resultat von Satz 18.6 kann in polynomieller Zeit derandomisiert werden.

Zusammenfassung: Bis 0.87 ist es also möglich, das Problem in polynomzeit zu lösen. Ab 0.97 nicht mehr. Was ist dazwischen? Woran liegt dieser Komplexitätssprung? Was passiert in diesem Bereich? Vielleicht werden wir es irgendwann einmal verstehen...

19 Dichte Subgraphen

Problem (Dense Subgraph). Zu einem Graphen $G = (V, E)$ und $k \in \mathbb{N}$ wird ein $V' \subseteq V$ mit $|V'| = k$ gesucht, so dass $|E(G[V'])|$ maximal ist.

Wir können auch eine gewichtete Version mit einer Gewichtsfunktion $w: E \rightarrow \mathbb{Q}_0^+$ betrachten.

Bekanntes:

1. Das Problem Dense Subgraph ist NP-hart.
2. Falls die Kantengewichte die Dreiecksungleichung erfüllen, d.h. $w_{i,j} \leq w_{i,k} + w_{k,j}$ für alle i, k, j , dann gibt es einen Approximationsalgorithmus mit Faktor $\frac{1}{2}$. (Hassin, Rubinstein, Taurir 1997)
3. Im allgemeinen gibt es einen Approximationsalgorithmus mit Lösung $\geq r \cdot \text{OPT}$, wobei $\frac{1}{r} = \tilde{O}(n^{0.3885})$. Hierbei steht \tilde{O} für die bekannte \mathcal{O} -Notation, wobei logarithmische Terme weggelassen werden. (Kortsarz, Releg 1993)

Betrachte $k = c \cdot n$, $c > 0$ Konstante, z.B. $k = \frac{n}{2}$. Aber auch hier ist das Problem NP-hart.

Modellierung als Semidefinites Programm: Seien x_1, \dots, x_n $-1/+1$ -Variablen und $S = \{i \in V \mid x_i = 1\}$. Wie kann man mit einer quadratischen Form

beschreiben, dass ein Knoten $j \in S$ ist?

Antwort: Man führt einen Dummyknoten 0 mit Variable x_0 ein. Wir sagen $i \in S \Leftrightarrow x_0x_i = 1$.

Problem (DS).

$$\begin{aligned} \max \quad & \frac{1}{4} \sum_{(i,j) \in E} w_{i,j} (1 + x_0x_i)(1 + x_0x_j) \\ & \sum_{i=1}^n x_0x_i = 2k - n \\ & x_0, x_1, \dots, x_n \in \{-1, 1\} \end{aligned}$$

Zwei Fälle $x_0x_i = 1$ und $x_0x_i = -1$: Sei l die Anzahl der i , so dass $x_i = 1$ und l' die Anzahl der i , so dass $x_i = -1$. dann gilt $l + l' = 2k - n$. Also $l = 2k - n - l'$. Also $l = k \Leftrightarrow k - n - l' = 0$. Also $l' = -(n - k)$.

Es folgt, dass (DS) äquivalent zu Dense Subgraph ist.

Problem (RDS).

$$\begin{aligned} \max \quad & \frac{1}{4} \sum_{(i,j) \in E} w_{i,j} (1 + \vec{x}_0 \vec{x}_i + \vec{x}_0 \vec{x}_j + \vec{x}_i \vec{x}_j) \\ & \sum_{i=1}^n \vec{x}_0 \vec{x}_i = 2k - n \\ & \vec{x}_0, \vec{x}_1, \dots, \vec{x}_n \in \mathbb{R}^{n+1} \\ & \|\vec{x}_i\|_2 = 1 \text{ für } i = 0, \dots, n \end{aligned}$$

Mit $y_{i,j} = \vec{x}_i \cdot \vec{x}_j$ erhalten wir

Problem (RDS1).

$$\begin{aligned} \max \quad & \frac{1}{4} \sum_{(i,j) \in E} w_{i,j} (1 + y_{0,i} + y_{0,j} + y_{i,j}) \\ & \sum_{i=1}^n y_{0,i} = 2k - n \\ & y_{i,i} = 1 \text{ für } i = 0, \dots, n \\ & Y = (y_{i,j}) \text{ symmetrisch und positiv definit.} \end{aligned}$$

Lemma 19.1. (RDS) ist äquivalent zu (RDS1) und (RDS1) ist ein semidefinites Programm, dass sich (bis auf jedes $\epsilon > 0$) in polynomieller Zeit in n und $\log \frac{1}{\epsilon}$ lösen lässt.

Algorithmus Dense Subgraph (DS)

1. Relaxierung: Löse (RDS1) und finde auf der Choleskyzerlegung von Y Vektoren $v_0, \dots, v_n \in S_{n+1}$.
2. Randomisiertes Runden: Wähle $r_t \in \mathbb{R}^{n+1}$, $\|r_t\| = 1$ zufällig unter Gleichverteilung und setze $S_t = \{i \in V \mid \text{sign}(v_i \cdot r_t) = \text{sign}(v_0 \cdot r_t)\}$.
3. Reparatur:
 - Falls $|S_t| < k$, füge beliebige $k - |S_t|$ Knoten zu S_t hinzu.
 - Falls $|S_t| > k$, entferne $i \in S_t$ mit minimum gewichteten Grad $\sum_{j \in S_t} w_{i,j}$.

Wiederhole bis $|S_t| = k$.

4. Iteration: Sei $T = T(\epsilon)$, $\epsilon > 0$. Wiederhole Schritt 2 und 3 für $t = 1, \dots, T$ und gebe die beste der Lösungen dieser T Läufe aus.

[von Katja Wolf, Srivastav 1998; Analyse G. Jäger, Srivastav 2005]

Lemma 19.2. Sei \tilde{S}_t die Menge nach Reparatur. Es gilt

$$w(\tilde{S}_t) \geq \frac{k(k-1)}{|S_t|(|S_t|-1)} w(S_t).$$

Beweis. Sei $i \in S_t$. Wenn wir die Kantengewichte des Subgraphen auf $S_t \setminus \{i\}$ aufsummieren, wird jede Kante $|S_t| - 2$ mal gezählt. Also gilt

$$\sum_{i \in S_t} w(S_t \setminus \{i\}) = (|S_t| - 2)w(S_t).$$

Sei v der erste Knoten, der entfernt wird. Dann ist

$$w(S_t \setminus \{v\}) \geq \frac{\sum_{i \in S_t} w(S_t \setminus \{i\})}{|S_t|} = \frac{(|S_t| - 2)w(S_t)}{|S_t|}.$$

Die Aussage beweist man nun durch Induktion. □

Lemma 19.3. Sei OPT^* der Wert einer optimalen Lösung von (RDS). Für $t = 1, \dots, T$ gilt

(i) $\mathbb{E}[w(S_t)] \geq \beta \cdot \text{OPT}^*$.

(ii) $\alpha k \leq \mathbb{E}[|S_t|] \leq (1 - \alpha)n + \alpha k$.

Mit $\alpha > 0.87856$ und $\beta > 0.79607$.

Beweis.

$$\mathbb{E}[w(S_t)] = \sum_{(i,j) \in E} w_{i,j} \Pr[\underbrace{\text{sign}(v_i r_t)}_a = \underbrace{\text{sign}(v_j r_t)}_b = \underbrace{\text{sign}(v_0 r_t)}_c].$$

Es gibt folgende Ereignisse:

- $A: a = b = c$
- $B: a \neq b = c$
- $C: a = c \neq b$
- $D: a = b \neq c$

Es gilt: $\Pr[A] + \Pr[B] + \Pr[C] + \Pr[D] = 1$. Wir können Gleichungen für die W'keiten bestimmen:

$$\Pr[B] = \Pr[b = c] - \Pr[A]$$

Analog für $\Pr[C]$ und $\Pr[D]$. Diese ergeben ein System von Gleichungen, dessen Auflösung nach $\Pr[A]$ ergibt:

$$\begin{aligned} \Pr[A] &= 1 - \frac{1}{2\pi} (\arccos(v_0 v_i) + \arccos(v_0 v_j) + \arccos(v_i v_j)) \\ &\geq \frac{\beta}{4} (1 + v_0 v_i + v_0 v_j + v_i v_j). \end{aligned}$$

Es folgt, dass

$$\mathbb{E}[w(S_t)] \geq \beta \cdot \text{OPT}^*.$$

Des weiteren gilt

$$\begin{aligned} \mathbb{E}[|S_t|] &= \sum_{i=1}^n \Pr[\text{sign}(v_i r_t) = \text{sign}(v_0 r_t)] = \sum_{i=1}^n \left(1 - \frac{1}{\pi} \arccos(v_0 v_i)\right) \\ &= n - \sum_{i=1}^n \frac{1}{\pi} \arccos(v_0 v_i) \leq n - \alpha \sum_{i=1}^n \frac{1 - v_0 v_i}{2} = n - \alpha n + \alpha k. \end{aligned}$$

□

Fixiere $c > 1$.

Lemma 19.4. *Sei $\epsilon > 0$ klein genug. Dann gibt es $p'(\epsilon) < 1$, so dass*

$$\Pr[|S_t| \notin [(\alpha - \epsilon)/c, n]] \leq p'(\epsilon).$$

Beweis. Übung. □

Indem wir das Zufallsexperiment $T' = T'(\epsilon)$ mal wiederholen, stellen wir sicher, dass es mit W'keit $(1 - \epsilon)$ eine Lösung S_τ gibt mit

$$|S_\tau| \in [(\alpha - \epsilon)n/c, n].$$

(Mache T' so groß, dass $p'(\epsilon)^{T'} \leq 1 - \epsilon$)

Satz 19.5. Sei $k = \frac{n}{c}$, dann berechnet der Algorithmus einen Subgraphen S derart, dass

$$\mathbb{E}[w(S)] \geq (1 - \frac{c}{n})(1 - \epsilon)^2 \hat{r}(c) \text{OPT},$$

wobei

$$\hat{r}(c) := \frac{\beta}{c^2} + \frac{(c^2 - 1)(c - 1)(1 - \epsilon)\alpha\beta}{c^4(c - \alpha + \epsilon) - c^2(c^2 - 1)(c - 1)(1 - \epsilon)\alpha}.$$

Beweis. $X_t := w(S_t)$, $Y_t := n - |S_t|$, $t = 1, \dots, T$.

$$Z_t := \frac{X_t}{f \cdot \text{OPT}^*} + \frac{Y_t}{n - \frac{n}{c}}.$$

$f > 0$ ist eine Konstante, die später bestimmt wird (in Abhängigkeit von $c, \alpha, \beta, \epsilon$). Lemma 19.3 liefert

$$\mathbb{E}[Z_t] \geq \underbrace{\frac{\beta}{f}}_{=:L} + \alpha. \tag{*}$$

Ein zufälliger Subgraph R mit $\frac{n}{c}$ Knoten hat erwartetes Gewicht

$$\mathbb{E}[w(R)] = \sum_{\{i,j\} \in E} w_{i,j} \underbrace{\Pr[i, j \in R]}_{1/c^2} = \frac{1}{c^2} w(V).$$

Also gibt es einen Subgraphen R mit $w(R) \geq \frac{1}{c^2} w(V)$. Es folgt

$$\text{OPT}^* \geq \text{OPT} \geq \frac{1}{c^2} w(V)$$

und damit

$$Z_t \leq \frac{w(V)}{f \frac{1}{c^2} w(V)} + \frac{n - |S_t|}{n - \frac{n}{c}} \leq \underbrace{\frac{c^2}{f} + \frac{c}{c-1}}_{=:M}. \tag{**}$$

(*) und (***) ergeben

$$\Pr[Z_t \leq (1 - \epsilon)\left(\frac{\beta}{f} + \alpha\right)] \stackrel{(1)}{\leq} \frac{M - L}{M - (1 - \epsilon)L} < 1.$$

zu (1):

$$\begin{aligned} L &\leq \mathbb{E}[Z_t] = \sum_{r \geq 0} r \cdot \Pr[Z_t = r] = \sum_{0 \leq r \leq M} r \Pr[Z_t = r] \\ &= \sum_{0 \leq r \leq (1 - \epsilon)L} r \Pr[Z_t = r] + \sum_{(1 - \epsilon)L < r \leq M} r \Pr[Z_t = r] \\ &\leq (1 - \epsilon)L \sum_{0 \leq r \leq (1 - \epsilon)L} \Pr[Z_t = r] + M \sum_{(1 - \epsilon)L < r \leq M} \Pr[Z_t = r] \\ &= (1 - \epsilon)L \cdot \underbrace{\Pr[Z_t \leq (1 - \epsilon)L]}_{=: p} + M(1 - p) \\ &= (1 - \epsilon)Lp + M - Mp = ((1 - \epsilon)L - M)p + M, \end{aligned}$$

also

$$L - M \leq ((1 - \epsilon)L - M)p$$

und damit

$$\frac{L - M}{(1 - \epsilon)L - M} \geq p.$$

Wiederhole das Experiment $T''(\epsilon)$ mal, so dass mit W'keit $1 - \epsilon$ für ein $\tau \leq T''(\epsilon)$ gilt

$$Z_\tau \geq (1 - \epsilon)\left(\frac{\beta}{f} - \alpha\right)$$

und

$$|S_t| \in \left[(\alpha - \epsilon)\frac{n}{c}, n\right].$$

Wähle λ und $\mu \in [(\alpha - \epsilon)/c, 1]$ so, dass $X_t = \lambda \cdot \text{OPT}^*$ und $|S_\tau| = \mu n$. Dann haben wir:

$$Z_\tau = \frac{\lambda}{f} + \frac{n - \beta n}{n - \frac{n}{c}} = \frac{\lambda}{f} + \frac{c}{c - 1}(1 - \mu)$$

$$\lambda \geq (1 - \epsilon)(\beta + \alpha f) - \frac{fc}{c - 1}(1 - \mu). \quad (***)$$

Fall 1. $\mu \in [(\alpha - \epsilon)/c, \frac{1}{c}]$.

$$\begin{aligned} w(\tilde{S}_t) &\geq w(S_\tau) = \lambda \text{OPT}^* \stackrel{(***)}{\geq} ((1 - \epsilon)(\beta + \alpha f) - \frac{fc}{c-1}(1 - \mu)) \text{OPT}^* \\ &\geq (1 - \frac{c}{n}) \underbrace{((1 - \epsilon)(\beta + \alpha f) - \frac{f}{c-1}(c - \alpha + \epsilon))}_{=:a} \text{OPT}^*. \end{aligned}$$

Fall 2. $\mu \in [\frac{1}{c}, 1]$. Mit Lemma 19.2 (?) folgt

$$w(\tilde{S}_\tau) \geq \frac{\frac{n}{c}(\frac{n}{c} - 1)}{\mu n(\mu n - 1)} w(S_\tau) \geq (1 - \frac{c}{n}) \frac{\lambda}{c^2 \mu^2} \text{OPT}^*.$$

$$\begin{aligned} \min_{\mu \in [\frac{1}{c}, 1]} \frac{\lambda}{c^2 \mu^2} &\stackrel{(***)}{\geq} \min_{\mu \in [\frac{1}{c}, 1]} \frac{(1 - \epsilon)(\beta + \alpha f) - \frac{fc}{c-1}(1 - \mu)}{c^2 \mu^2} \\ &\stackrel{\text{Kurvendiskussion}}{=} \min\{\underbrace{(1 - \epsilon)(\beta + \alpha f) - f}_{=:b}, \underbrace{(1 - \epsilon)(\beta + \alpha f)c^{-2}}_{=:b'}\}. \end{aligned}$$

Man zeigt, dass $a \leq b$ gilt. Damit interessiert und

$$\min\{a, b'\}.$$

Wir wählen nun f so, dass dieses Minimum möglichst groß wird. $a = a(f)$, $b' = b'(f)$ sind linear fallende bzw. steigende Funktionen. Bestimme also f so, dass $a(f) = b'(f)$. Man erhält

$$f = \frac{(c^2 - 1)(c - 1)(1 - \epsilon)\beta}{c^2(c - \alpha + \epsilon) - (c^2 - 1)(c - 1)(1 - \epsilon)\alpha}.$$

Insgesamt folgt:

$$w(\tilde{S}_\tau) \geq (1 - \frac{c}{n})b'(f) \cdot \text{OPT}^* \geq (1 - \frac{c}{n})(1 - \epsilon)\hat{r}(c)\text{OPT}.$$

Da dies mit W'keit $(1 - \epsilon)$ gilt, erhalten wir die Behauptung. \square

20 Probabilistisch verifizierbare Beweise und Nicht-Approximierbarkeit

20.1 Die Klasse $\text{PCP}(r, q)$

Sei Π ein endliches Entscheidungsproblem. Für eine Instanz $x \in \Pi$ sei $\mathcal{L}(x)$ die (evtl. leere) Menge der Lösungen. Wir fragen, ob x eine Lösung besitzt.

Definition 20.1. Ein *polynomiell zeitbeschränkter Verifier* V ist eine nicht deterministische Turingmaschine, die zu einer Eingabe $x \in \Pi$ und einem Beweis τ , $x, \tau \in \Sigma^*$ mit Σ endliches Alphabet für das Problem, entscheidet, ob x eine Lösung besitzt oder nicht. D.h. $V: \Pi \times \Sigma^* \rightarrow \{0, 1\}$.

Definition 20.2 (NP). NP ist die Klasse von endlichen Problemen, so dass folgendes gilt: Für jedes solche Problem Π gibt es einen polynomiell zeitbeschränkten Verifier V , so dass

- (i) für alle $x \in \Pi$, x hat eine Lösung gibt es $\tau_0 \in \Sigma^*$, $|\tau_0| = \text{poly}(|x|)$ mit $V(x, \tau_0) = 1 = \text{“Ja”}$ und
- (ii) für alle $x \in \Pi$, x hat *keine* Lösung gilt $V(x, \tau) = 0 = \text{“Nein”}$ für alle $\tau \in \Sigma^*$.

Wir erweitern den Begriff des Verifiers: V ist nun eine Funktion $V: \Pi \times \Sigma^* \times \Sigma^* \rightarrow \{0, 1\}$, wobei $x \in \Pi$ eine Eingabe des Problems, $\tau \in \Sigma^*$ ein Beweis und $\rho \in \Sigma^*$ eine zufällige Folge in Σ^* ist und $V(x, \tau, \rho)$ die Ja/Nein-Antwort auf die Frage, ob x eine Lösung hat, ist.

Definition 20.3 (Die Klasse $\text{PCP}(r, q)$). Sei $0 < \epsilon < 1$. $\text{PCP}_\epsilon(r(n), q(n))$ ist die Klasse von endlichen Problemen, für die es einen $(r(n), q(n))$ -zeitbeschränkten Verifier $V: \Pi \times \Sigma^* \times \Sigma \rightarrow \{0, 1\}$ gibt, so dass

- (i) für alle $x \in \Pi$, x hat eine Lösung gibt es ein $\tau \in \Sigma^*$, so dass $\Pr_\rho[V(x, \tau, \rho) = \text{“Ja”}] = 1$.
- (ii) für alle $x \in \Pi$, x hat *keine* Lösung gilt $\Pr_\rho[V(x, \tau, \rho) = \text{“Nein”}] \geq 1 - \epsilon$ für alle $\tau \in \Sigma^*$.

Dabei sind $r, q: \mathbb{N} \rightarrow \mathbb{N}$, der Verifier V liest x, τ der Länge $\mathcal{O}(q(n))$, ρ der Länge $\mathcal{O}(r(n))$ und gibt $V(x, \tau, \rho)$ in Zeit polynomiell in $|x|$ aus.

Setze $\text{PCP}(r(n), q(n)) := \text{PCP}_{\frac{1}{2}}(r(n), q(n))$.

\Pr_ρ bedeutet die W'keit bezüglich der Verteilung, die ρ zugrunde liegt.

Proposition 20.4. *Es gilt*

$$\text{PCP}(0, 0) = P \text{ und } \text{PCP}(0, \text{poly}(n)) = NP.$$

Proposition 20.5. *Es gilt*

- (i) $PCP(r(n), q(n)) \subseteq NTIME(\text{poly}(n) \cdot 2^{\mathcal{O}(r(n))})$,
- (ii) $PCP(\text{poly}(n), \text{poly}(n)) \subseteq NEXP$,
- (iii) $PCP(\log n, \text{poly}(n)) = NP$ und
- (iv) $PCP(\log n, 1) \subseteq NP$.

Beweis. zu (i) Sei $x \in \Pi$. Aus V konstruiere eine nicht-deterministische TM (NDTM) M wie folgt: Für jedes $\rho \in \Sigma^*$ lasse $V(x, \tau, \rho)$ laufen, wobei τ irgendein Beweis ist. Die Anzahl der Läufe ist $\mathcal{O}(2^{\mathcal{O}(r(n))})$. Jeder Lauf ist in $\mathcal{O}(\text{poly}(n))$ Zeit möglich, also erhalten wir eine Komplexität von $\mathcal{O}(\text{poly}(n) \cdot 2^{\mathcal{O}(r(n))})$.

M gibt genau dann "Ja" aus, wenn für jedes ρ , $V(x, \tau, \rho) = \text{"Ja"}$.

Für M gilt dann: Falls x eine Lösung hat, so ist $M(x, \tau) = \text{"Ja"}$ falls für jedes ρ , $V(x, \tau, \rho) = \text{"Ja"}$. Falls x keine Lösung hat, so ist $M(x, \tau) = \text{"Nein"}$ falls ein ρ existiert mit $V(x, \tau, \rho) = \text{"Nein"}$.

Also ist M die gesuchte NDTM.

zu (ii) $PCP(\text{poly}(n), \text{poly}(n)) \stackrel{(i)}{\subseteq} NTIME(\text{poly}(n)2^{\text{poly}(n)}) = NTIME(2^{\text{poly}(n)}) \stackrel{\text{Def.}}{=} NEXP$.

zu (iii) Benutze (i) und $NTIME(\text{poly}(n)) = NP$.

zu (iv) Benutze (i) und $NTIME(\text{poly}(n)) = NP$. □

Satz 20.6 (Babai, Fortnow, Lund '80er Jahre). *Es gilt*

$$PCP(\text{poly}(n), \text{poly}(n)) = NEXP.$$

Satz 20.7 (Arora, Lund, Motwani, Sudan, Szegdy 1992). *Es gilt*

$$PCP(\log n, 1) = NP.$$

Cliquen-Problem

Zu einem Graph $G = (V, E)$ wird eine Clique maximaler Kardinalität gesucht. Sei $\omega(G)$ die *Cliquenzahl*, d.h. die Größe einer Clique mit maximaler Kardinalität. Die Berechnung dieser Zahl ist NP-schwer. Für die Approximierbarkeit gilt folgendes Resultat:

Satz 20.8. *Es gibt keinen polynomiellen 2-Faktor Approximationsalgorithmus für die Cliquenzahl, es sei denn $P = NP$.*

Der Satz gilt sogar für einen Approximationsfaktor von $n^{1-\epsilon}$, für jedes $0 < \epsilon < 1$.

Beweis. Sei F eine 3-SAT-Formel mit n Variablen und m Klauseln. Wir konstruieren einen Graphen G_F mit den Eigenschaften:

- a) wenn F erfüllbar ist, dann gilt $\omega(G_F) = f(n)$ und
- b) wenn F nicht erfüllbar ist, dann gilt $\omega(G_F) < \frac{1}{2}f(n)$,

für eine geeignete Funktion $f(n)$, die später bestimmt wird.

Das PCP-Theorem 20.7 besagt, dass $NP = PCP(\log n, 1)$. $3\text{-SAT} \in NP$, also gibt es einen $(r(n), q(n))$ -beschränkten Verifier V mit $r(n) = \mathcal{O}(\log n)$ und $q(n) = \mathcal{O}(1)$. Sei π ein Beweis für F . Wir nehmen der Einfachheit halber an, dass $\pi = (\pi_1, \pi_2, \dots)$, $\pi_i \in \{0, 1\}$. Der Verifier V testet für einen zufälligen Vektor τ der Länge $r(n)$, $q(n)$ Einträge von π , sagen wir die Variablen $x_{i_1}, \dots, x_{i_{q(n)}}$ und gibt eine Variablenbelegung $a_{i_1}, \dots, a_{i_{q(n)}}$ aus. Sei G_F der folgende Graph, $G_F = (V, E)$:

Für einen Beweis π sei V_π die Menge aller Tupel $(\tau, a_{i_1}, \dots, a_{i_{q(n)}})$, $|\tau| \leq r(n)$. Der $(r(n), q(n))$ -Verifier liest gewisse Bits $i_1, \dots, i_{q(n)}$ des Beweises π , erhält die Bits $a_{i_1}, \dots, a_{i_{q(n)}}$ und akzeptiert. Zwei Tupel $(\tau, a_{i_1}, \dots, a_{i_{q(n)}})$ und $(\tau', b_{j_1}, \dots, b_{j_{q(n)}})$ sind durch eine Kante verbunden, wenn:

- $\tau \neq \tau'$,
- entweder alle Variablen $x_{i_1}, \dots, x_{i_{q(n)}}$ und $x_{j_1}, \dots, x_{j_{q(n)}}$ sind verschieden oder
- die Variablen, die in beiden gemeinsam sind haben die gleiche Belegung, d.h. aus $x_{i_k} = x_{j_l}$ folgt $a_{i_k} = b_{j_l}$.

Das definiert den Graphen $G_{F,\pi}$ mit der Knotenmenge V_π . Wir definieren dann

$$G_F := \bigcup_{\pi \text{ Beweis für } F} G_{F,\pi}.$$

Behauptung: Für $f(n) := 2^{r(n)}$ gilt

$$\omega(G_F) = f(n) \Pr_{\tau} [V(F, \tau, \pi) = 1].$$

Beweis: Wir zeigen dazu

(i) $\omega(G_F) \leq f(n) \Pr_\tau[V(F, \tau, \pi) = 1]$ und

(ii) $\omega(G_F) \geq f(n) \Pr_\tau[V(F, \tau, \pi) = 1]$.

zu (i) Sei C eine Clique in G_F mit $|C| = \omega(G_F)$. Dann ist C eine Clique in einem gewissen G_{F, π_0} . Dann ist $|\leq| \#(\tau, a_{i_1}, \dots, a_{i_{q(n)}})$ die V akzeptiert $\leq 2^{r(n)} \Pr_\tau[V(F, \tau, \pi_0) = 1]$.

zu (ii) Cliquen in G_F sind auch welche im entsprechenden $G_{F, \pi}$. Sei C die Menge aller Tupel $(\tau, a_{i_1}, \dots, a_{i_{q(n)}})$ die V akzeptiert. Dann ist C eine Clique und $\omega(G_F) \geq |C| = 2^{r(n)} \Pr_\tau[V(F, \tau, \pi) = 1]$.

Damit ist die Behauptung gezeigt.

Hieraus folgen a) und b):

zu a) Ist F erfüllbar, dann gibt es einen Beweis π_0 mit $\Pr_\tau[V(F, \tau, \pi_0) = 1] = 1$.

zu b) Ist F nicht erfüllbar, dann gilt für alle Beweise π : $\Pr_\tau[V(F, \tau, \pi) = 1] < \frac{1}{2}$.

Angenommen es gibt einen polynomiellen 2-Faktor-Approximationsalgorithmus für die Cliquenzahl. Dieser gibt eine Lösung $c(G_F)$ aus, mit

$$c(G_F) \geq \frac{1}{2} \cdot \omega(G_F) \begin{cases} = \frac{1}{2} f(n) & F \text{ erfüllbar} \\ < \frac{1}{4} f(n) & F \text{ nicht erfüllbar} \end{cases} .$$

Falls also $c(G_F) \geq \frac{1}{2} f(n)$ ist, so gilt natürlich auch $\omega(G_F) \geq \frac{1}{2} f(n)$. Mit b) folgt, dass F erfüllbar ist.

Falls $c(G_F) < \frac{1}{2} f(n)$ ist, so folgt F ist nicht erfüllbar.

Damit hätten wir einen polynomiellen Algorithmus für 3-SAT, d.h. $P = NP$. \square

21 Färben 3-färbbarer Graphen mit semidefiniter Relaxierung

Wir betrachten immer Graphen $G = (V, E)$ mit $V = \{1, \dots, n\}$ und $|E| = m$.

Definition 21.1. Sei $k \in \mathbb{N}$.

(i) Eine k -Färbung von G ist eine Abbildung $f: V \rightarrow \{1, \dots, k\}$ mit $f(i) \neq f(j)$ für alle $\{i, j\} \in E$.

- (ii) G heißt k -färbbar, wenn es eine k -Färbung von G gibt. “Färbe G mit k Farben” bedeutet: finde eine k -Färbung von G .
- (iii) Die *chromatische Zahl* $\chi(G)$ ist die kleinste natürliche Zahl k so, dass G k -färbbar ist.

Das Bestimmen von $\chi(G)$ ist für beliebige Graphen G NP-hart.

Bemerkung. Falls $\chi(G) = 2$ ist, so kann eine 2-Färbung in polynomieller Zeit gefunden werden. Auch der Test, ob $\chi(G) = 2$ oder $\chi(G) \neq 2$ kann polynomiell erfolgen.

Satz 21.2 (von Brooks). *Man findet eine Δ -Färbung in polynomieller Zeit, wobei $\Delta = \max_{v \in V} \deg(v)$. Ausnahmen: K_n (brauche $n = \Delta + 1$ Farben) und C_n , n ungerade (brauche 3 Farben).*

Beweisskizze. Sei $v_0 \in V$, $\deg(v_0) < \Delta$. Bilde Schichten mit Knoten, gleichen Abstands zu v_0 . Färbe dann die Äusserste Schicht mit so wenig Farben wie möglich. Dazu braucht man maximal $\Delta - 1$ Farben.

Dies klappt nur für nicht reguläre Graphen. Für reguläre Graphen, ausser K_n und C_n , n ungerade, lässt es sich auch zeigen, ist aber wesentlich aufwendiger. \square

Algorithmus von Wigderson

Eingabe: G , $\delta \geq 0$.

Ausgabe: Färbung von G .

1. Wähle Knoten $v \in V$ mit $\deg(v) > \delta$. Gibt es keinen solchen, verwende Satz von Brooks.
2. a) Färbe v mit neuer Farbe.
b) Färbe die Nachbarn von v ($N(v)$) mit “möglichst wenig” Farben (z.B. optimal).

Setze $G := G \setminus N(v) \setminus \{v\}$.

3. Gehe zu 1.

Satz 21.3. *Für 3-färbbaren Graphen G liefert der Algorithmus mit $\delta := \sqrt{n}$ eine $\mathcal{O}(\sqrt{n})$ -Färbung.*

Dabei können die Nachbarn von v optimal gefärbt werden, da G 3-färbbar ist und wir also eine 2-Färbung der Nachbarn suchen.

Relaxierung zur Vektorfärbung

Definition 21.4. (i) Eine *Vektorfärbung* (VF) von G ist eine Abbildung $V \rightarrow \mathbb{R}^n$, $i \mapsto v_i$, wobei $\|v_i\|_2 = 1$ für alle $i \in V$.

(ii) Sei $k > 1$ eine reelle Zahl. Eine Vektorfärbung v_1, \dots, v_n heißt eine *k-Vektorfärbung*, wenn für alle $\{i, j\} \in E$ gilt $v_i \cdot v_j \leq -\frac{1}{k-1}$.

Proposition 21.5. (i) Sei $k \in \{2, \dots, n+1\}$. Dann gibt es k Einheitsvektoren $v_1, \dots, v_k \in \mathbb{R}^n$, d.h. $\|v_i\|_2 = 1$, so dass $v_i \cdot v_j = -\frac{1}{k-1}$ für alle $i, j \in \{1, \dots, k\}$, $i \neq j$.

(ii) Jeder k -färbbare Graph hat eine k -Vektorfärbung.

Definition 21.6. Eine *k-Semifärbung* von G besteht aus einer Menge $U \subseteq V$ und einer Abbildung $f: U \rightarrow \{1, \dots, k\}$, so dass $|U| \geq \frac{n}{2}$ und f eine Färbung auf $G[U]$ (der durch U induzierte Subgraph von G) ist.

Idee für einen Algorithmus

Eingabe: Ein Graph G , 3-färbbar und $\epsilon' \in (0, 1)$.

1. Bestimme eine $(3 + \epsilon)$ -Vektorfärbung für geeignetes $\epsilon = \epsilon(\epsilon')$ in Abhängigkeit von ϵ' .
2. Runde diese zu einer Semifärbung mit $\mathcal{O}(\Delta^{\log_3 - \epsilon'})$ Farben.
3. Ergänze die Semifärbung um "wenige" Farben zu einer Färbung von G .

Eine mögliche Verbesserung: Reduziere zunächst Δ mit Wigdesons Technik.

Konstruktion einer $(k + \epsilon)$ -Vektorfärbung

Definition 21.7. Eine *k-Matrixfärbung* von G ist eine symmetrische positiv semidefinite $(n \times n)$ -Matrix M , für die gilt $M_{i,i} = 1$ und $M_{i,j} \leq -\frac{1}{k-1}$ falls $\{i, j\} \in E$.

Proposition 21.8. (i) G hat eine k -Vektorfärbung genau dann, wenn G eine k -Matrixfärbung hat.

(ii) Zu gegebener k -Matrixfärbung kann in Zeit polynomiell in k , n und $\log(\frac{1}{\epsilon})$ eine $(k + \epsilon)$ -Vektorfärbung konstruiert werden.

Beweis. (i) „ \Rightarrow “ Setze $M_{i,j} := v_i \cdot v_j$ und rechne nach.

„ \Leftarrow “ Da M symmetrisch positiv semidefinit ist findet man eine obere Dreiecksmatrix U , so dass $M = U^T U$ (Cholesky-Zerlegung). Die Spalten von U liefern die Vektorfärbung.

(ii) Führe eine approximative Cholesky-Zerlegung durch.

□

Wie erhalten wir die k -Matrixfärbung?

Problem (SDP).

$$\begin{aligned} & \min \alpha \\ & M \text{ pos. semidef.} \\ & M_{i,j} \leq \alpha \text{ für alle } \{i,j\} \in E \\ & M_{i,i} = 1, \quad M_{i,j} = M_{j,i} \text{ für alle } i,j \in V \end{aligned}$$

Das Optimum von (SDP) kann bis auf einen gewünschten Fehler δ approximiert werden in Zeit polynomiell in n und $\log \frac{1}{\delta}$. Für eine geeignete Wahl von δ in Abhängigkeit von ϵ kann somit zuerst eine $k + 2\delta(k-1)^2$ -Matrixfärbung und daraus eine $(k+\epsilon)$ -Vektorfärbung berechnet werden. (Falls G k -färbbar.)

Runden der $(3 + \epsilon)$ -Vektorfärbung

1. Wähle zufällig und unabhängig r Hyperebenen in \mathbb{R}^n (zufällig heißt hier: Wähle einen Normalenvektor aus der Einheitssphäre unter Gleichverteilung). r hängt von Δ und ϵ' ab.
2. Die Hyperebenen teilen \mathbb{R}^n in 2^r Bereiche R_1, \dots, R_{2^r} . Färbe alle Knoten, deren Vektor der Vektorfärbung in R_s liegt mit der Farbe s , $s = 1, \dots, 2^r$.
3. Lösche die Farbe auf ungünstig gefärbten Knoten.

Satz 21.9. *Mit Wahrscheinlichkeit $\geq \frac{1}{2}$ liegt nach dieser Prozedur eine $\mathcal{O}(\Delta^{\log_{3-\epsilon'} 2})$ -Semifärbung vor (wenn ϵ geeignet gewählt wird).*