Probabilistische Algorithmen.

Algorithmus für weighted Vertex Cover:

$E[m_{pr}(x)] \leq 2m^*(x)$. Sei $U^* \in S^*(x)$.

Zufallsvariable $X_v := \begin{cases} w(v) & v \in U \\ 0 & \text{sonst.} \end{cases}$

$$E[m_{pr}(x)] = E[\sum_{v \in U} w_v] = E[\sum_{v \in U} X_v]$$
$$= E[\sum_{v \in V} X_v] = \sum_{v \in V} E[X_v]$$

und

$$m^*(x) = \sum_{v \in U^*} w(v) \geq \sum_{v \in U^*} E[X_v].$$

Also z.z.

$$\sum_{v \in V} E[X_v] \leq 2 \sum_{v \in U^*} E[X_v].$$

Für Kante $e = \{u, v\}$: e wählt v wenn
e gewählt ⇝ im nächsten Schritt v

Zufallsvariable $X_{e,v} := \begin{cases} w(v) & \text{e wählt v} \\ 0 & \text{sonst} \end{cases}$

Ist $X_{e,v} = w(v)$, dann $X_{e',v} = 0$ für alle $e' \neq e$ mit $v \in e'$

$\rightsquigarrow X_v = \sum_{e \ni v} X_{e,v}$

$\rightsquigarrow E[X_v] = \sum_{e \ni v} E[X_{e,v}]$

Für $e = \{u,v\}$ ist

$$E[X_{e,v}] = E[X_{e,u}] \quad , \text{ denn}$$

$$\begin{aligned}
E[X_{e,v}] &= w(v)\cdot \mathbb{P}[e \text{ wählt } v] \\
&= w(v)\cdot \mathbb{P}[e \text{ gewählt}]\cdot \frac{w(u)}{w(u)+w(v)} \\
&= w(u)\cdot \mathbb{P}[e \text{ gewählt}]\cdot \frac{w(v)}{w(u)+w(v)} \\
&= w(u)\cdot \mathbb{P}[e \text{ wählt } u] \\
&= E[X_{e,u}].
\end{aligned}$$

Also: $$\sum_{v\in U^*} E[X_v] = \sum_{v\in U^*}\sum_{e\ni v} E[X_{e,v}] \qquad (*)$$

und

(**) $$\sum_{v\notin U^*} E[X_v] = \sum_{v\notin U^*}\sum_{e\ni v} E[X_{e,v}] = \sum_{v\notin U^*}\sum_{\substack{e\ni v\\ e=\{u,v\}}} E[X_{e,u}]$$

Für $v\notin U^*$ ist $u\in U^*$ für alle $u\in N(v)$.

$\rightsquigarrow$ Für jeden Summanden $E(X_{e,u})$ in (**) ein gleicher in (*)

$$\rightsquigarrow \quad \sum_{v\notin U^*} E[X_v] \le \sum_{v\in U^*} E[X_v]$$

Also:

$$\sum_{v\in V} E[X_v] = \sum_{v\in U^*} E[X_v] + \sum_{v\notin U^*} E[X_v] \le 2\sum_{v\in U^*} E[X_v]$$

$\square$

Einfacher SAT-Algo

jede Variable $\alpha(x) = \begin{cases} 0 & \text{W'keit } \frac{1}{2} \\ 1 & \text{"} \end{cases}$ faire Münze.

Für $C = a_1 \vee \dots \vee a_k$ gilt

$$\Pr[\alpha \models C] = 1 - \frac{1}{2^k}$$

Also $E[m(F,\alpha)] = \sum_{C \in F} \Pr[\alpha \models C]$

also falls $|C| \geq k$ für alle $C \in F$, dann

$$E[m(F,\alpha)] \geq m\left(1 - \frac{1}{2^k}\right).$$

$\rightsquigarrow$ $E[m(F,\alpha)] \geq \frac{1}{2} m$ $\quad | \boxed{2\text{-Approx}}$

$\geq \frac{3}{4} m$ alle Klauseln ≥ 2 Literale

$\geq \frac{7}{8} m$ $\quad \geq 3$ Literale ...

Analyse von LP-SAT

$$E[m(F,\alpha)] = \sum_{C_j \in F} \Pr[\alpha \models C_j]$$

$$= \sum_{C_j \in F} \left(1 - \prod_{x_i \in C_j} (1 - y_i^*) \prod_{\bar{x}_i \in C_j} y_i^*\right)$$

Lemma: $\geq \sum_{C_j \in F} \alpha_k z_j^* \geq \alpha_k \sum z_j^*$

$= \alpha_k \, m^*(LP(F)) \geq \alpha_k \, m^*(F)$ □

Beweis des Lemmas

OBdA $C_j = x_{j1} \vee \ldots \vee x_{jk}$ positiv.

also z.z.:

$$1 - \prod_{i=1}^{k} (1 - y^*_{j,i}) \geq a_k z_j^*$$

$$1 - \prod_{i=1}^{k} (1 - y^*_{j,i}) \geq 1 - \left(\frac{\sum_{i=1}^{k} (1 - y^*_{j,i})}{k} \right)^k$$

$$= 1 - \left(\frac{k - \sum_{i=1}^{k} y^*_{j,i}}{k} \right)^k$$

$$= 1 - \left(1 - \frac{\sum_{i=1}^{k} y^*_{j,i}}{k} \right)^k$$

$$= 1 - \left(1 - \frac{z_j^*}{k} \right)^k \geq a_k z_j^*$$

da $\lambda y.\ 1 - (1 - \frac{y}{k})^k$ ist konkav auf $[0,1]$

und $\geq$ gilt für $y = 0$ und $y = 1$:

$$\Leftarrow \quad 1 - \left(1 - \frac{0}{k}\right)^k \geq a_k \cdot 0 \quad \checkmark$$

$$1 - \left(1 - \frac{1}{k}\right)^k \geq a_k \cdot 1 \quad \checkmark \qquad \square$$

Also: Performance für $k=2$ ~ $\frac{4}{3}$

für $k \geq 3$ ... max. 1.582 ...

$\gamma_1 = \frac{1}{2}$ $a_1 = 1$ $\leadsto$ $\gamma_1 + a_1 = \frac{3}{2}$

$\gamma_2 = \frac{3}{4}$ $a_2 = \frac{3}{4}$ $\leadsto$ $\gamma_2 + a_2 = \frac{3}{2}$

$e = 2.7183$
$\frac{1}{e} = 0.3679$
$1 - \frac{1}{e} = 0.6321$

für $k \geq 3$ ist $\gamma_k \geq \frac{7}{8}$ und $a_k \geq 1 - \frac{1}{e}$

$$\gamma_k + a_k \geq \frac{7}{8} + \left(1 - \frac{1}{e}\right) \geq \frac{3}{2}$$

Also

$$\frac{W_1 + W_2}{2} \geq \sum_{k \geq 1} \sum_{C_j \in F_k} \frac{3}{4} z_j^* = \frac{3}{4} \sum_{k \geq 1} \sum_{C_j \in F_k} z_j^*$$

$$\geq \frac{3}{4} \sum_{C_j \in F} z_j^*$$

$$= \frac{3}{4} M^*(LP(F))$$

$\square$

Also: Performanz von KombiSAT ist mind. $\boxed{\frac{4}{3}}$

Analyse von KombiSAT:

$$W_1 := E[m(F, \alpha_1)]$$
$$W_2 := E[m(F, \alpha_2)]$$

$$E[m(F, \alpha)] = \max(W_1, W_2)$$

z.Z.: $\max(W_1, W_2) \geq \frac{3}{4} m^*(F)$.

Es gilt $\max(W_1, W_2) \geq W_1 + W_2 / 2$

und $m^*(F) \leq m^*(LP(F))$

also reicht z.Z. $\frac{W_1 + W_2}{2} \geq \frac{3}{4} m^*(LP(F))$.

Sei $\gamma_k := (1 - \frac{1}{2^k})$. $F_k := \{ C \in F, \ |C| = k \}$

$$W_1 \geq \sum_{k \geq 1} \sum_{C_j \in F_k} \gamma_k \geq \sum_{k \geq 1} \sum_{C_j \in F_k} \gamma_k z_j^* . \qquad (*)$$

nach Lemma

$$W_2 \geq \sum_{k \geq 1} \sum_{C_j \in F_k} a_k z_j^* \qquad (**)$$

(*) + (**):

$$\frac{W_1 + W_2}{2} \geq \sum_{k \geq 1} \sum_{C_j \in F_k} \frac{\gamma_k + a_k}{2} z_j^*$$

Symmetrische Matrizen, positiv semidefinit.

(2) → (1)

Sei λ Eigenwert von A, also $Av = \lambda v$ für ein $v \in \mathbb{R}^n$.

Dann ist $0 \leq v^T A v = \lambda v^T v$.

Da $v^T v \geq 0$, folgt $\lambda \geq 0$ $\square$

(1) → (3)

$\lambda_1, \dots, \lambda_n$ Eigenwerte $v_1, \dots, v_n$ Eigenvektoren Orthonormalbasis.

$$Q := (v_1 \dots v_n) \qquad \Lambda := \begin{pmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_n \end{pmatrix}$$

$Av_i = \lambda_i$ f. alle i, also $AQ = Q\Lambda$.

Q orthogonal $\rightsquigarrow$ $QQ^T = I$ $\rightsquigarrow$ $Q^T = Q^{-1}$.

also. $$A = Q\Lambda Q^T$$

$$D := \begin{pmatrix} \sqrt{\lambda_1} & & 0 \\ & \ddots & \\ 0 & & \sqrt{\lambda_n} \end{pmatrix} \rightsquigarrow \Lambda = DD^T$$

also $$A = QDD^TQ^T = (QD)(QD)^T$$ $\square$

(3) → (2) Sei $x \in \mathbb{R}^n$.

$$x^T A x = x^T W^T W x = (Wx)^T (Wx) \geq 0$$

$\square$

Konstruktion im Beweis (1)-(3) braucht Eigenwerte $\lambda_1 \dots \lambda_n$, (SPEKTRUM) dies ist exponentiell zu berechnen ..

Gibt aber besseren Algorithmus :

Cholesky:

Rekursion nach Dimension:

$\dim M = 1$ $\quad M = (a)$ $\quad a \geq 0$ $\quad \rightsquigarrow \sqrt{a}$ ok.

$$(\sqrt{a})(\sqrt{a}) = (a) \quad \checkmark$$

---

ist $M = \begin{pmatrix} a & q^T \\ q & N \end{pmatrix}$,

dann ist $N - \frac{1}{a} q q^T \geq 0$. (oB.)

Sei V mit $V^T V = N - \frac{1}{a} q q^T$.

$$\begin{pmatrix} \sqrt{a} & 0^T \\ \frac{1}{\sqrt{a}} q & V^T \end{pmatrix} \begin{pmatrix} \sqrt{a} & \frac{1}{\sqrt{a}} q^T \\ 0 & V \end{pmatrix}$$

$$= \begin{pmatrix} a & q^T \\ q & \frac{1}{a} q q^T + V^T V \end{pmatrix}$$

$$= \begin{pmatrix} a & q^T \\ q & \frac{1}{a} q q^T + N - \frac{1}{a} q q^T \end{pmatrix}$$

$$= \begin{pmatrix} a & q^T \\ q & N \end{pmatrix}$$

$\square$

Semidefinite Programmierung

Verallgemeinerung von LP.

Alle Constraints sind linear, außer

$X \succeq 0$. $\triangleq$ unendlich viele lineare Constraints $y^T X y \geq 0 \quad \forall y \in \mathbb{R}^n$.

Exakte Lösung nicht effizient möglich (oder überhaupt nicht.)

aber beliebig gute Näherung!

Laufzeit abhängig von $|P|$ und $\frac{1}{\varepsilon}$ (Näherung) und r! In allen Anwendungen r (Norm der Lösungen) klein.

MAXIMUM CUT

$\rightsquigarrow$ Lokale Suche

Analyse des Algorithmus:

Betrachte $e = \{u, v\}$

$$\Pr[e \text{ durchbrennt}] = \Pr[u \in V_0 \wedge v \in V_1] + \Pr[u \in V_1 \wedge v \in V_0]$$
$$= \frac{1}{4} + \frac{1}{4} = \frac{1}{2}.$$

$$E[m_{Pr}] = \sum_{e \in E} \Pr[e \text{ durchbrennt}] = \sum_{e \in E} \frac{1}{2} = \frac{1}{2}|E| \geq \frac{1}{2} m^* .$$

$\square$

Quadratisches Programm

$y_i \in \{\pm 1\} \quad \hat{=} \quad y_i^2 = 1$

$\{v_i, v_j\}$ durchtrennt $\quad y_i y_j = -1$

nicht $\quad y_i y_j = 1$

$$m \quad \frac{1}{2} \sum_{\{v_i v_j\}} (1 - y_i y_j)$$

Vektorprogramm

ersetze $y_i \in \mathbb{Z}$ durch $z_i \in \mathbb{R}^n$

Produkt $y_i y_j \rightsquigarrow$ Skalarprodukt $z_i^T z_j$

$y_i^2 = 1 \rightsquigarrow \|z_i\| = 1$ (Einheitsvektor)

$y_1 \dots y_n$ Lösung für Max Cut

$\rightsquigarrow \quad z_i := \begin{pmatrix} y_i \\ 0 \\ \vdots \\ 0 \end{pmatrix}$ Lösung für Vek(G)

$z_i^T z_j = y_i y_j$, also

$m(z_1 \dots z_n) = m(y_1 \dots y_n)$

Optimale Lsg $y_1^* \dots y_n^*$ — Lösung mit gleichem Wert für Vek(G)

Optimum für Vek(G) höchstens besser!

$$m^*(\text{Vek}(G)) \geq m^*(G).$$

SDP(G) äquivalent zu Vek(G)

Sei $z_1, \dots, z_n$ Lösung von Vek(G).

Matrix $X = (x_{ij})$ definiert durch $x_{ij} = z_i^T z_j$,

also $Z = (z_1 \dots z_n)$ $\leadsto$ $X = Z^T Z$.

$\leadsto$ Lösung von SDP(G), da

$x_{ii} = 1$, da $\|z_i\| = 1$

X pos. semidefinit nach Satz $\square$

Umgekehrt

X Lösung von SDP(G), Cholesky-Faktorisierung $X = Z^T Z$

$Z = (z_1 \dots z_n)$ z_i Spaltenvektoren von Z.

$z_i^T z_j = x_{ij}$ $\leadsto$ Lösung von Vek(G), da

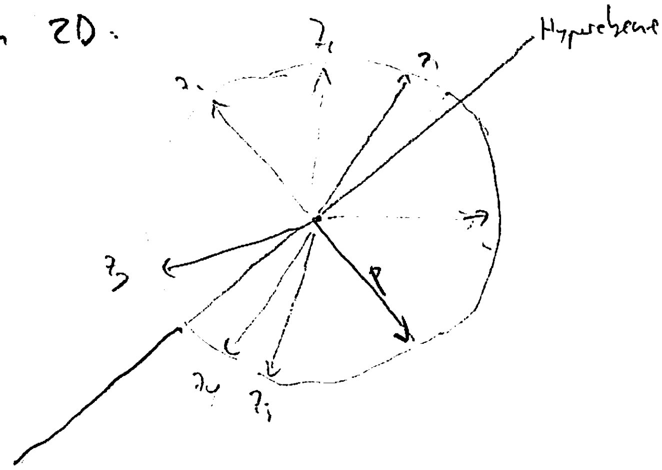
$\|z_i\| = 1$, da $x_{ii} = z_i^T z_i = 1$ $\square$

Algorithmus von G.-W.

Aus Lösung $z_1, \dots, z_n$ von Vek(G)

Lösung von MAX CUT für G.

In 2D:



$z_1 \dots z_n$ Vektoren auf der $n-1$-dimensionalen Einheitskugel

Teile durch Hyperebene H 2 Halbkugeln

Hyperebene $H \mathrel{\hat{=}}$ Orthogonaler Vektor $\vec{p}$

oberhalb von H $\leftrightarrow$ $p^T z_i \geq 0$

unterhalb $\leftrightarrow$ $p^T z_i < 0$

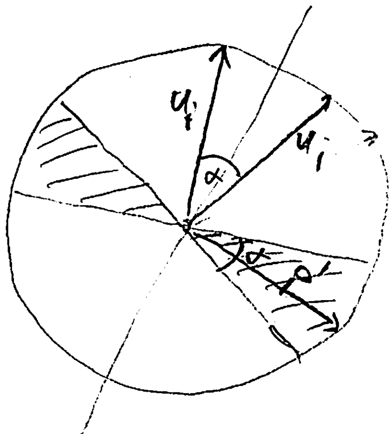
$\leadsto$ Schnitt

Analyse:

Die Wahrscheinlichkeit, dass x_i x_j in verschieden Mengen liegen, also $\{x_i, x_j\}$ getrennt wird, ist

$$\frac{1}{\pi} \arccos(u_i^T u_j)$$

□



$\cos(\alpha) = u_i^T u_j$, also

$$\alpha := \arccos(u_i^T u_j)$$

p' = Projektion von p auf $\mathrm{span}(u_i, u_j)$

$$p^T u_i = p'^T u_i \quad \text{und} \quad p^T u_j = p'^T u_j.$$

liefern verschiedene VZ, wenn p' in grauem Segment.

p gleichverteilt in S^{n-1} $\leadsto$ p' gleichverteilt (Rotation).

$$\leadsto \text{Wahrscheinlichkeit} = \frac{2\alpha}{2\pi} = \frac{1}{\pi} \arccos u_i^T u_j \quad \checkmark$$

□

Lemma $\forall z \in [-1, 1]$ $\quad \frac{1}{\pi} \arccos z \geq 0.87856 \, \frac{1-z}{2}$.

Also

$$E(m_{\text{par}}) = \sum_{\{v_i, v_j\} \in E} \mathbb{P}[\{v_i, v_j\} \text{ durchtrennt}]$$

$$= \sum_{\{v_i, v_j\} \in E} \frac{1}{\pi} \arccos(u_i^T u_j)$$

$$\geq 0.878567 \sum_{\{v_i, v_j\} \in E} \frac{1}{2}(1 - u_i^T u_j)$$

für $\epsilon \leq 7 \cdot 10^{-6}$

$$\geq 0.878567(m^* - \epsilon) \geq 0.87856\, m^*$$

□