Max-Cut Problem $G = (V, E)$, $w : E \longrightarrow \mathbb{Q}$ 1

$A \subseteq E \Rightarrow w(A) \stackrel{def}{=} \sum_{e \in A} w(e)$

$V' \subseteq V \Rightarrow \{e \in E;\ |e \cap V'| = 1\}$ is <u>edge-cut</u>

Find edge-cut of max weight.

- A basic NP-complete problem [unweighted]
- $w : E \to \mathbb{Q}^{-} \Rightarrow$ polynomial [max flow - min cut]
- Polynomial for graphs embeddable on any fixed surface and the weights polynomially bounded (in $|G|$)
- Seminal approximation alg. by SemiDef. Programming

Structure of edge-cuts

$G = (V, E)$ graph

I_G incidence matrix (over $\mathbb{F}_2 = (\{0,1\}, \dot{+} \bmod 2)$)

$$\begin{array}{c|ccc} & & \{x,y\} = e & \\ & \cdots & 0 & \cdots \\ x & \cdots & 1 & \cdots \\ V & \cdots & 0 & \cdots \\ y & \cdots & 1 & \cdots \\ & \cdots & 0 & \cdots \end{array} \quad E$$

$A \subseteq E$ <u>even</u> if (V, A) has all degrees even

Cycle Space (Prostor Cyklů)

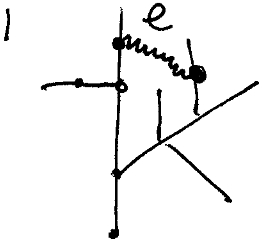
$$\mathcal{E}_G = \{A \subseteq E;\ A \text{ even}\}$$

Cut Space (Prostor Řezů)

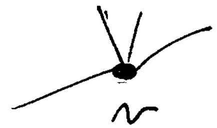
$$\mathcal{C}_G = \{A \subseteq E;\ A \text{ edge-cut}\}$$

- $\mathcal{E}_G = \mathrm{Ker}_2(I_G)$
- $\mathcal{C}_G = \mathrm{Im}_2(I_G)$
- $\dim \mathcal{E}_G = |E| - |V| + \#\text{components}$
- $\dim \mathcal{C}_G = |V| - \#\text{components}$

basis of $\mathcal{E}_G$: T : spanning tree of G, $e \in E \setminus T$,
C_e : unique cycle of $T \cup \{e\}$.
$\mathcal{B}_T = \{ C_e \,;\, e \in E \setminus T \}$.

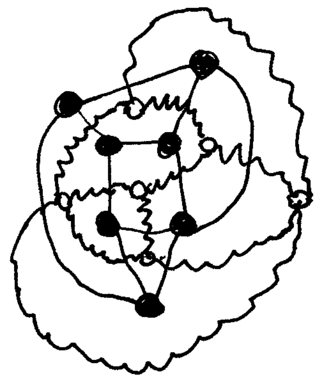


basis of $\mathcal{C}_G$: $\mathcal{B}_0 = \{ \{ e \,;\, v \in e \} \,;\, v \in V \}$



G planar, G^* geometric dual $\Rightarrow$

$$\mathcal{C}_G \simeq \mathcal{E}_{G^*}$$

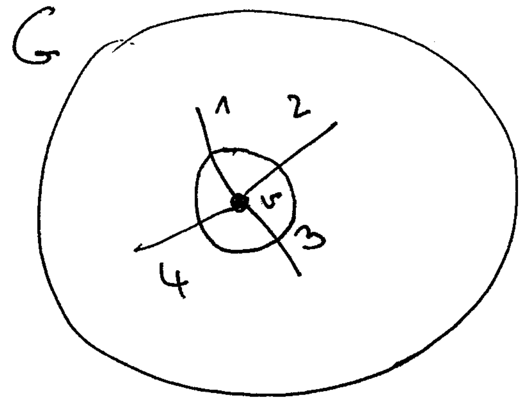


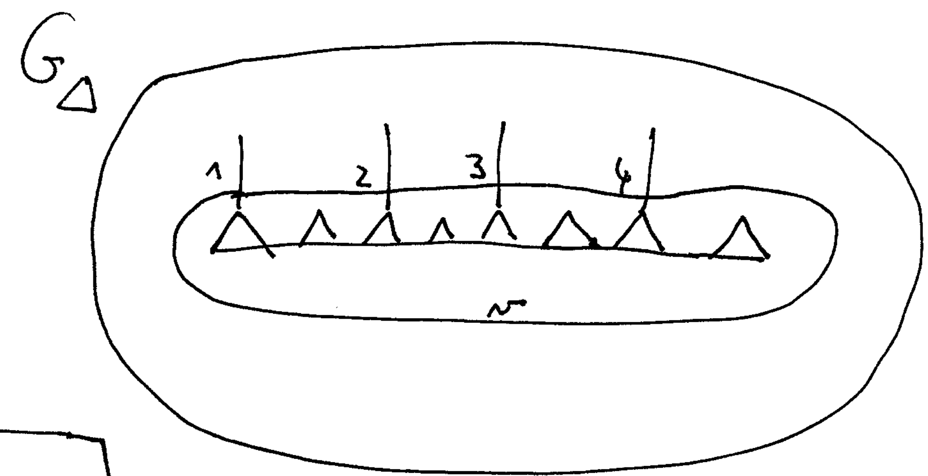
— G
~~~ G^*

G planar $\Rightarrow$ $\mathcal{E}_G$ generated by face-boundaries
~~~

- Max-Cut NP-complete
- Max Even Set Polynomial

Reduction to weighted perfect matching [Fischer 60 physicist]





<u>Thm</u>. $A \subseteq E$ even iff A can be uniquely extended to a perfect matching of G_Δ by the gadget edges.

<u>Proof</u>: simple

weights of the gadget edges $\equiv 0$

All other weights unchanged.

Why was a physisist Fisher interested in perfect matchings, edge-cuts (in 1960)?

Statistical Physics

perfect matching = dimer arrangement

edge-cut = state of Ising model

<u>40's Ising</u>: understanding of Ising model on planar graphs

<u>50's, 60's</u>: Attempts to understand Onsager's result combinatorially: Kac, Ward, Feynmen, Fischer, Kastelyn, Sherman ...

Ising Model on $G=(V,E)$

<u>state</u>: $s : V \to \{1,-1\}$ spin

<u>energy</u>:

$$E(s) = \sum_{\{i,j\}\in E} + s_i s_j w(ij)$$

Each state defines edge-cut: $\{e;\ s(i) \neq s(j)\}$

$C(s)$

$$w(C(s)) = [-E(s) + w(E)]/2 \quad (*)$$

Semidefinite Programming Algorithm for Max-Cut

$G = (V, E)$; $V = \{1, \dots, n\}$

- Variables $x_1, \dots, x_n \in \{1, -1\}$
- Assignment of values from $\{1, -1\}$ to $x_1, \dots, x_n$ encodes a cut $(S, V \setminus S)$, where $S = \{i \in V;\ x_i = 1\}$.
- $\frac{1 - x_i x_j}{2}$ is the contribution of edge $\{i,j\}$ to this cut.
- $\boxed{\text{Max Cut}}$ $Opt(G)$: $\max \sum_{ij \in E} \frac{1 - x_i x_j}{2}$; $x_i \in \{1, -1\}$, $i = 1, \dots, n$.
- Replace x_i by $w_i \in S^{n-1} = \{x \in \mathbb{R}^n;\ \|x\| = 1\}$

$$\max \sum_{ij \in E} \frac{1 - w_i^T w_j}{2} \ ; \quad w_i \in S^{n-1}$$ is $\boxed{\text{upper bound}}$ to $Opt\,G$.

$\boxed{\text{vector program}}$

(*) Semidefinite Programming solves efficiently to any desired accuracy

<u>Substitute</u>: $x_{ij} := u_i^T u_j$

$\boxed{SDP(G)}$ $\quad \text{Max} \sum_{\{i,j\} \in E} \frac{1 - x_{ij}}{2}$; $x_{ii} = 1$, $i = 1, 2, \dots, n$

$X \succcurlyeq 0$ positive semidefinite

$X = U^T U$, the columns of U, $u_1, \dots, u_n$ form feasible solution of the vector program.

$M \in Sym_n$ [symmetric matrix] ~~....~~ The following is equivalent

1. M <u>positive semidefinite</u>, i.e., all eigenvalues non-negative
2. $x^T M x \geq 0$ for all $x \in \mathbb{R}^n$
3. There is $U \in \mathbb{R}^{n \times n}$, $M = U^T U$ [Cholesky Factorization]

Columns of U unit vectors iff $X_{ii} = 1$, $i = 1, \dots, n$

(*) We can find in polynomial time matrix $X^* \succcurlyeq 0$, $X^*_{ii} = 1$, $i = 1, \dots, n$ s.t. $\sum_{ij \in E} \frac{1 - X^*_{ij}}{2} \geq SDP(G) - \varepsilon$, for every $\varepsilon > 0$.

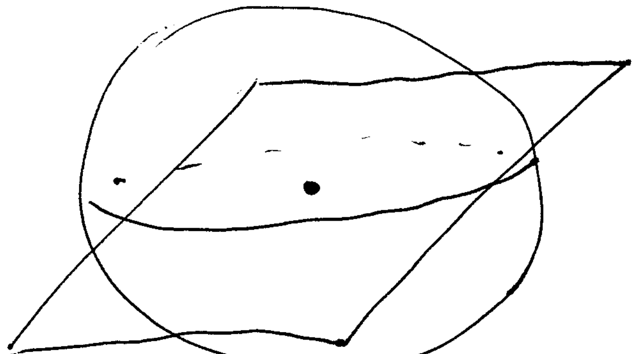
(*) We compute U^* s.t. $X^* = (U^*)^T U^*$ up to a tiny error.

<u>Hence</u> (U^* has columns $u_1^*, u_2^*, \ldots, u_n^*$)

$$\sum_{\{i,j\}\in E} \frac{1-u_i^{*T} u_j^*}{2} \geq SDP(G) - \varepsilon \geq Opt(G) - \varepsilon .$$

Rounding the Vector Solution

We want to map S^{n-1} back to S^0 so that we do not lose too much.



cut by a random hyperplane

randomised rounding

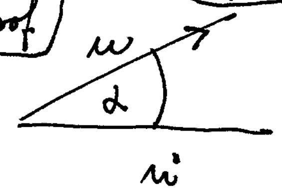
- We take p uniformly at random from S^{n-1}

$$u \longrightarrow \begin{cases} 1 & \text{if } p^T u \geq 0 \\ -1 & \text{otherwise} \end{cases}$$

- $u, u' \in S^{n-1}$. Probability that u, u' get different values is

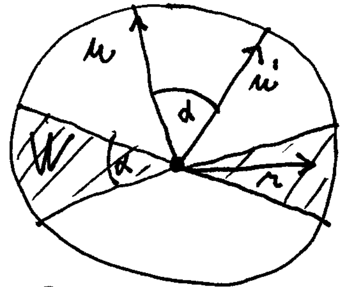
$$\frac{1}{\pi} \arccos u^T u'$$

Proof



$\alpha = \arccos u^T u' \in [0, \pi]$

since $\cos\alpha = u^T u' \in [-1, 1]$



- u, u' go differently iff projection r of p lies in W. r uniformly distributed in $[0, 2\pi]$

$\Rightarrow$ Probability is $\frac{\alpha}{\pi}$. ⊠

The expected number of edges in the resulting cut equals

$$\sum_{\{i,j\}\in E} \frac{\arccos u_i^{*T} u_j^*}{\pi}$$

$\boxed{\text{Lemma}}$ For all $z \in [-1,1]$ $\quad \dfrac{\arccos(z)}{\pi} \geq 0{,}8785672 \, \dfrac{1-z}{2}$

$\boxed{\text{Hence}}$

$$\sum_{\{i,j\}\in E} \frac{\arccos u_i^{*T} u_j^*}{\pi} \geq 0{,}8785672 \sum_{\{i,j\}\in E} \frac{1 - u_i^{*T} u_j^*}{2} \geq 0{,}8785672\left(\mathrm{Opt}(G) - \varepsilon\right)$$

$$\geq 0{,}878\, \mathrm{Opt}(G)$$

if $\varepsilon \leq 5 \cdot 10^{-4}$.

Approximation Algorithms P optimization problem (maximization)

- $\underline{I}$: set of instances ; $i \in \underline{I} \Rightarrow F(i)$ set of feasible solutions
- $s \in F(i)$ has a non-negative real value $w(s) \geq 0$.
- $Opt(i) = \sup_{s \in F(i)} w(s) \in \mathbb{R}_+ \cup \{-\infty, \infty\}$.

- A algorithm to solve P is Δ-approximation algorithm for P if
  $[\Delta : \mathbb{N} \longrightarrow \mathbb{R}_+]$
  - there is polynomial p s.t. for all $i \in \underline{I}$, the runtime of A on i is $\leq p(|i|)$
  - $i \in \underline{I} \Rightarrow w(A(i)) \geq \Delta(|i|) \cdot Opt(i)$

$\boxed{\text{Randomised algorithms}}$ A randomised Δ-approximation algorithm has $\boxed{\text{expected}}$ poly runtime and must satisfy $\mathbb{E}[w(A(i))] \geq \Delta(|i|) \, Opt(i)$
[$w(A(i))$ is random variable]

A randomized 0,5- approximation algorithm for Max-Cut

$G=(V,E)$; Pick S a random subset of V [each vertex $v \in V$ included in S with probability $1/2$, independent of all other vertices.

$\boxed{\text{Thm}}$ This is a randomised 0,5-approximation algorithm for Max Cut.

$\boxed{\text{Proof}}$ $\mathbb{E}[w(A(G))] = \mathbb{E}[|E(S, V \setminus S)|] = \sum_{e \in E} \text{Prob}[e \in E(S, V \setminus S)] =$

$$\sum_{e \in E} \frac{1}{2} = \frac{1}{2}|E| \geq \frac{1}{2} Opt(G).$$

$e \in E(S, V \setminus S)$ iff exactly one of the two end-points of e ends up in S.

$\boxed{LP}$

$$\max c^T x$$
$$Ax = b$$
$$x \geq 0$$

$c \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, $A \in \mathbb{R}^{m \times n}$

X <u>feasible</u> solution if $X \in SYM_n$, $A(X) = b$, $X \succeq 0$.

<u>The value</u> is

$$\sup\{C \bullet X : A(X) = b, X \succeq 0\}$$

Semidefinite Programming

$SYM_n = \{X \in \mathbb{R}^{n \times n} ; x_{ij} = x_{ji}, 1 \leq i < j \leq n\}$ [replaces $x \in \mathbb{R}$]

$A : SYM_n \to \mathbb{R}^m$ linear [replaces $A \in \mathbb{R}^{m \times n}$]

$$X \bullet Y := \sum_{i=1}^{n} \sum_{j=1}^{n} x_{ij} y_{ij}, \quad X, Y \in SYM_n$$

● $X \bullet Y = Tr(X^T Y)$

$X \succeq 0$ (being positive semidefinite) [replaces $x \geq 0$]

$$\begin{aligned} &\text{Max } C \bullet X \\ &\left.\begin{aligned} A_1 \bullet X &= b_1 \\ &\vdots \\ A_m \bullet X &= b_m \end{aligned}\right\} \\ &X \succeq 0 \end{aligned}$$

$A(X) = b$

where

$b = (b_1, \dots, b_m)$

$A : SYM_n \to \mathbb{R}^m$ linear

$$A = (a_{ijk})_{i,j=1}^{n} \Big|_{k=1}^{m}$$

WARNINGS

1. Finite value is not necessarily attained:

$$\operatorname{Max} -x_{11}$$
$$x_{12} = 1$$
$$X \succeq 0$$

Feasible solutions: all $X \succeq 0$, $X = \begin{pmatrix} x_{11} & 1 \\ 1 & x_{22} \end{pmatrix}$

$$\Updownarrow$$

$$x_{11} \geq 0,\ x_{22} \geq 0,\ x_{11} x_{22} \geq 1$$

$$\Updownarrow$$

$$x_{11} > 0 \text{ and } x_{22} \geq \frac{1}{x_{11}}$$

Value 0 but no opt. solution

---

* All known eff. algorithms return only $\boxed{\text{aproximately optimal solutions}}$ [ellipsoid method]

---

* Cholesky factorisation also can be constructed only aproximately